

Patent  
Attorney Docket No. 030681-457

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of

In-kyu PARK et al

Application No.: 10/727,629

Filing Date: December 5, 2003

Title: METHOD AND APPARATUS FOR ENCODING AND DECODING THREE-DIMENSIONAL OBJECT DATA

Group Art Unit: 2671

Examiner: Unassigned

Confirmation No.: 9103

SUBMISSION OF CERTIFIED COPY OF PRIORITY DOCUMENT

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

The benefit of the filing date of the following priority foreign application(s) in the following foreign country is hereby requested, and the right of priority provided in 35 U.S.C. § 119 is hereby claimed.

Country: Republic of Korea

Patent Application No(s): 2003-83778

Filed: November 24, 2003

In support of this claim, enclosed is a certified copy(ies) of said foreign application(s). Said prior foreign application(s) is referred to in the oath or declaration. Acknowledgment of receipt of the certified copy(ies) is requested.

Respectfully submitted,

BURNS, DOANE, SWECKER & MATHIS, L.L.P.

P.O. Box 1404  
Alexandria, Virginia 22313-1404  
(703) 836-6620  
Date: April 15, 2004

By

Charles F. Wieland III

Registration No. 33,096



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto  
is a true copy from the records of the Korean Intellectual  
Property Office.

출 원 번 호 : 10-2003-0083778  
Application Number

출 원 년 월 일 : 2003년 11월 24일  
Date of Application NOV 24, 2003

2006

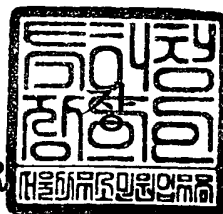
출 원 인 : 삼성전자주식회사 외 1명  
Applicant(s) SAMSUNG ELECTRONICS CO., LTD., et al.



2003 년 12 월 02 일

특 허 청

COMMISSIONER



## 【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0030
【제출일자】	2003.11.24
【국제특허분류】	H04N
【발명의 명칭】	3 차원 객체 데이터 부호화 및 복호화 방법 및 장치
【발명의 영문명칭】	Method and apparatus for encoding and decoding 3 dimensional data
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【출원인】	
【명칭】	재단법인 서울대학교산학협력재단
【출원인코드】	2-2003-007067-6
【대리인】	
【성명】	이영필
【대리인코드】	9-1998-000334-6
【포괄위임등록번호】	2003-003435-0
【대리인】	
【성명】	이해영
【대리인코드】	9-1999-000227-4
【포괄위임등록번호】	2003-003436-7
【발명자】	
【성명의 국문표기】	박인규
【성명의 영문표기】	PARK, In Kyu
【주민등록번호】	720228-1249123
【우편번호】	156-773
【주소】	서울특별시 동작구 사당2동 우성아파트 208동 1505호
【국적】	KR
【발명자】	
【성명의 국문표기】	이신준
【성명의 영문표기】	LEE, Shin Jun

【주민등록번호】	731103-1953212
【우편번호】	137-130
【주소】	서울특별시 서초구 양재동 10-32번지 303호
【국적】	KR
【발명자】	
【성명의 국문표기】	송인욱
【성명의 영문표기】	SONG, In Wook
【주민등록번호】	760620-1024519
【우편번호】	151-866
【주소】	서울특별시 관악구 신림2동 408-120 303호
【국적】	KR
【발명자】	
【성명의 국문표기】	이상욱
【성명의 영문표기】	LEE, Sang Uk
【주민등록번호】	490811-1005912
【우편번호】	135-904
【주소】	서울특별시 강남구 압구정2동 한양아파트 43동 403호
【국적】	KR
【발명자】	
【성명의 국문표기】	김창수
【성명의 영문표기】	KIM, Chang Su
【주민등록번호】	711112-1051710
【우편번호】	153-764
【주소】	서울특별시 금천구 시흥2동 벽산아파트 522동 1305호
【국적】	KR
【우선권주장】	
【출원국명】	US
【출원종류】	특허
【출원번호】	60/430,976
【출원일자】	2002. 12. 05
【증명서류】	미첨부

## 【우선권 주장】

## 【출원국명】

US

## 【출원종류】

특허

## 【출원번호】

60/487,263

## 【출원일자】

2003.07.16

## 【증명서류】

미첨부

## 【심사청구】

청구

## 【취지】

특허법 제42조의 규정에 의한 출원; 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인

이영필 (인) 대리인

이해영 (인)

## 【수수료】

## 【기본출원료】

20 면 29,000 원

## 【가산출원료】

52 면 52,000 원

## 【우선권주장료】

2 건 43,000 원

## 【심사청구료】

30 항 1,069,000 원

## 【합계】

1,193,000 원

## 【첨부서류】

1. 요약서·명세서(도면)\_1통 2. 위임장\_1통 3. 우선권증명서류 및 동 번역문\_2통

**【요약서】****【요약】**

본 발명은 포인트텍스처(PointTexture), 체적소(voxel), 팔진트리(octree) 구조의 데이터 중 어느 하나로 이루어진 3차원 객체 데이터 부호화 및 복호화 방법 및 장치에 관한 것으로서, 그 부호화 방법은 포인트텍스처, 체적소, 트리구조의 데이터 중 어느 하나로 이루어진 3차원 객체 데이터에 대해 노드의 종류를 나타내는 라벨이 부여된 트리구조의 3차원 객체 데이터를 생성하는 단계; 라벨이 부여된 트리구조의 3차원 객체 데이터의 노드를 부호화하는 단계; 및 노드 부호화된 데이터를 비트스트림으로 생성하는 단계를 포함한다. 그 부호화장치는 3차원 객체 데이터에 대해 노드마다 노드의 종류를 나타내는 라벨이 부착된 트리구조의 3차원 객체 데이터를 생성하는 트리구조 생성부; 트리구조의 3차원 객체 데이터에 대해 상기 라벨을 참조하여 병합하는 병합순서선택부; 병합된 노드를 부호화하는 노드부호화부; 및 노드 부호화된 데이터를 비트스트림으로 생성하는 비트스트림생성부를 포함한다. 그 복호화방법은 부호화된 3차원 객체 데이터의 비트스트림에 대해 노드 복호화하는 단계; 및 노드 복호화된 데이터를 트리구조로 복원하는 단계를 포함한다. 그 복호화장치는 부호화된 3차원 객체 데이터의 비트스트림을 받아들이는 비트스트림독출부; 비트스트림을 복호화하는 노드복호화부; 및 복호화된 노드를 트리구조로 복원하는 트리구조복원부를 포함한다.

**【대표도】**

도 11

**【명세서】****【발명의 명칭】**

3차원 객체 데이터 부호화 및 복호화 방법 및 장치{Method and apparatus for encoding and decoding 3 dimensional data}

**【도면의 간단한 설명】**

도 1은 계층화된 깊이 이미지에 대한 포인트텍스처의 예를 도시한 것이다.

도 2는 포인트텍스처 노드 규격을 도시한 것이다.

도 3은 라벨이 부착된 볼륨데이터의 팔진트리 표현을 도시하고 있다.

도 4는 상기 부호화 방법을 이용하여, 비트량-왜곡(rate-distortion) 곡선을 도시한 것이다.

도 5는 노드의 비트스트림의 구조를 도시한 것이다.

도 6은 'S'노드의 비트스트림 구조를 도시하고 있다.

도 7은 'P'노드의 비트스트림 구조를 도시한 것이다.

도 8은 콘텍스트의 예를 도시한 것이다.

도 9는 본 발명의 성능을 시험하기 위한 테스트 모델을 도시한 것이다.

도 10 도 포인트텍스처 테스트 모델에 대해 WinZip과 본 발명에 따른 방법의 압축성능을 비교한 것을 도시하고 있다.

도 11은 본 발명에 의한 3차원객체 데이터 부호화 장치를 블록도로 도시한 것이다.

도 12는 병합순서 선택부의 구성을 블록도로 도시한 것이다.

도 13 은 후보노드 선택부의 동작을 흐름도로 도시한 것이다.

도 14는 병합순서선택부의 동작을 흐름도로 도시한 것이다.

도 15는 노드부호화부의 구성을 도시한 것이다.

도 16은 노드부호화부의 동작을 흐름도로 도시한 것이다.

도 17은 S노드부호화부의 동작을 흐름도로 도시한 것이다.

도 18은 P노드부호화부의 동작을 흐름도로 도시한 것이다.

도 19는 3차원 객체 데이터의 복호화장치의 구성을 블록도로 도시한 것이다.

도 20은 노드복호화부의 구성을 블록도로 도시한 것이다.

도 21은 노드복호화부의 동작을 흐름도로 도시한 것이다.

도 22는 S노드복호화부의 동작을 흐름도로 도시한 것이다.

도 23은 P노드복호화부의 동작을 흐름도로 도시한 것이다.

#### 【발명의 상세한 설명】

#### 【발명의 목적】

#### 【발명이 속하는 기술분야 및 그 분야의 종래기술】

<24> 본 발명은 데이터 부호화 및 복호화 방법 및 장치에 관한 것으로서, 보다 상세하게는 포인트텍스처, 체적소(voxel), 팔진트리 구조의 데이터 중 어느 하나로 이루어진 3차원 객체 데이터 부호화 및 복호화 방법 및 장치에 관한 것이다.

<25> 3차원 그래픽에 관한 연구의 초기부터 연구자들의 궁극적인 목표는 실제 이미지와 같은 현실적인(realistic) 그래픽 화면을 생성하는 것이다. 따라서, 전통적인 렌더링 기술분야에서 다각형 모델(Polygonal model)을 이용하는 연구들이 수행



되어 왔고 그 결과로서 매우 현실적인(realistic) 3차원 환경을 제공하기에 충분할 정도로 모델링과 렌더링 기술이 개발되어 왔다. 그러나 복잡한 모델을 생성하기 위한 과정은 전문가의 많은 노력과 시간을 필요로 한다. 또한 현실적이고 복잡한 환경은 막대한 양의 정보를 필요로 하며, 저장 및 전송에 있어서 저효율을 초래한다.

<26> 현재 컴퓨터 그래픽에서 3차원 객체 표현의 주요한 방법은 다각형 모델이다. 임의의 형상을 색상 다각형의 집합, 즉 삼각형들에 의해 대략적으로 표현할 수 있다. 소프트웨어 알고리즘의 엄청난 진보와 그래픽 하드웨어의 발전으로 인해 복잡한 객체 및 장면을 실시간으로 상당히 현실적인 정지 및 동영상 다각형 모델로 시각화하는 것이 가능하다.

<27> 그러나 또 다른 3차원 표현에 대한 연구가 지난 수 년 동안 매우 활발하게 진행되어 왔다. 현실 세계의 객체들에 대한 다각형 모델을 구성하는 것이 어려울 뿐만 아니라 렌더링의 복잡성과 진정으로 사진과 같이 현실적인 장면을 생성하는 데 있어서의 품질이 만족스럽지 않다는 것이 이러한 연구가 진행되어 왔던 주요한 이유이다.

<28> 필요한 애플리케이션은 막대한 양의 다각형을 필요로 한다. 예를 들어, 인체의 상세한 모델은 수백만 개의 삼각형들을 포함하며, 이를 다루는 것은 용이한 일이 아니다. 비록 3차원 스캐너와 같이 3차원 측정 기술분야에서의 최근의 진보에 의해 수용가능한 에러를 갖는 조밀한 3차원 데이터를 얻을 수 있지만, 전체 객체에 대해 연속적으로 완벽한 다각형 모델을 얻는 것은 여전히 비용이 많이 들고 매우 어렵다. 다른 한편, 사진과 같이 현실적인 품질을 얻기 위한 렌더링 알고리즘은 연산에 있어 복잡하므로 결코 실시간 렌더링이 가능하지 않다.

<29> 깊이 이미지 기반 표현(Depth image-based representation : DIBR)은 복잡한 기하학적 구조를 가진 객체를 표현하고 렌더링하는 새로운 표현방법 및 렌더링 방법이다. DIBR 은 MPEG-4 AFX(Animation Framework eXtension)에 채택되어 왔다. 컴퓨터 그래픽에서는 전형적으

로 다각형 메쉬(polygonal mesh)로 객체(object)를 표현하여 왔지만, DIBR 은 3차원 객체를 눈에 보이는 객체의 표면을 감싸는(covering) 참조 이미지(reference image)의 집합으로 나타낸다. 각 참조 이미지는 상응하는 깊이 맵(depth map)으로 나타내어지며, 상기 깊이 맵은 이미지 평면의 픽셀로부터 객체 표면까지의 거리를 배열(array)로 나타낸 것이다. DIBR의 잇점 중 하나는 다각형모델을 직접 사용하지 않고도 참조 이미지가 객체에 대한 양질(high quality)의 시각화(visualization)를 제공할 수 있다는 것이다. 게다가, DIBR 뷰(view)를 렌더링하는 복잡도는 풍경(scene)의 복잡성에는 관계없이 뷰의 픽셀 수(즉 뷰의 해상도)에만 관련된다.

DIBR 패밀리의 주된 형식(format)은 심플 텍스처(SimpleTexture), 포인트 텍스처(PointTexture) 및 팔진트리 이미지(OctreeImage)이다. 상기 포인트텍스처는 하나의 카메라 위치에서 바라본 픽셀들의 배열로 객체를 표현한다. 각각의 포인트텍스처 픽셀은 색깔, 깊이(픽셀로부터 카메라까지의 거리) 및 포인트텍스처 렌더링을 보조하는 몇 개의 다른 속성들에 의해 표현된다. 각 시선(line of sight)을 따라 다수의 픽셀들이 있을 수 있으며, 따라서 포인트 텍스처는 보통 다수의 층(multiple layer)으로 구성된다. 1차원에서의 포인트텍스처의 간단한 예가 도 1에 도시되어 있다. 포인트텍스처는 전형적으로 대량의 데이터를 포함한다. 실제 이미지(realistic image)는 보다 높은 샘플링 밀도와 엄청난 양의 데이터를 요구한다. 그러므로 포인트텍스처 이미지의 압축은 효율적으로 수행되어야 한다. 포인트텍스처의 노드 규격(specification)은 도 2에 도시되어 있다. 도 2의 노드 규격에서 'depth'와 'color' 필드가 압축 대상이다.

<30>       포인트텍스처에 관한 이전의 연구는 거의 없다. Duan 과 Li 가 포인트텍스처 영상을 압축하는 방법을 제시하였는데, 상기 방법은 계층화된 깊이 이미지(Layered Depth Image:LDI)의

압축방법이라 불린다.[J.Duan J.Li Compression of the Layered Depth Image, IEEE Trans. Image Precessing, vol.12, no.3,pp.365-372, March 2003]

<31> 깊이정보를 압축하기 위해, JPEG-LS 알고리즘이 적용되었다. 또한 색깔(color)정보는 현존하는 코딩(coding) 표준을 사용하여 압축된다. 그러나 이러한 알고리즘은 점진적(progressive) 압축 및 전송을 지원하지 못한다. 패턴 코드 표현(PCR)을 사용하여 3차원 체적소(voxel) 표면 모델을 압축하는 방법이 제안되었다. [C.S. Kim and S.U.Lee, Compact Encoding of 3D Voxel Surface Based on Pattern Code Representation, IEEE Trans. Image Processing, vol. 11, no.8, pp.932-943,2002]

<32> 그러나 이 방법은 계층적 팔진트리(hierachical octree) 구조를 이용하지 아니하며, 점진적 압축 스킴(progressive compression scheme)을 용이하게 하는 것도 불가능하다. MPEG-4 AFX 표준에서, PPM(prediction by partial matching) 방법을 이용한 팔진트리 압축방법이 제안되었다.[ISO/IEC JTC1/SC29/WG11 14496-16:2003, Information Technology - Coding of Audio-Visual Objects - Part 16: Animation Framework eXtension (AFX)] 그러나 이것 역시 점진적 비트스트림(progressive bitstream)을 만들어내지는 못한다.

#### 【발명이 이루고자 하는 기술적 과제】

<33> 본 발명이 이루고자 하는 기술적 과제는 깊이 이미지 정보를 압축하기 위한 보다 효율적인 예측 부호화 및 높은 부호화 효율 및 효율적인 손실 압축을 제공하는, 포인트텍스처, 체적소, 팔진트리구조의 데이터 중 어느 하나로 이루어진 3차원 객체 데이터 부호화/복호화 방법 및 장치를 제공함에 있다.

## 【발명의 구성 및 작용】

<34> 상술한 기술적 과제를 달성하기 위한 본 발명에 의한, 3차원 객체 데이터 부호화 방법은, 포인트텍스처, 체적소, 팔진트리구조의 데이터 중 어느 하나로 이루어진 3차원 객체 데이터에 대해 노드의 종류를 나타내는 라벨이 부여된 트리구조의 3차원 객체 데이터를 생성하는 단계; 상기 라벨이 부여된 트리구조의 3차원 객체 데이터의 노드를 부호화하는 노드 부호화단계; 및 상기 노드 부호화된 데이터를 비트스트림으로 생성하는 비트스트림 생성단계를 포함함을 특징으로 한다. 상기 라벨이 부여된 트리구조의 3차원 객체 데이터 생성은 3차원 바운딩 볼륨을 사용하여 체적소 데이터로 변환하여 객체가 존재하는 위치의 체적소와 객체가 존재하지 않는 위치의 체적소를 라벨을 붙여 구분하여 표시함을 특징으로 한다. 상기 라벨을 붙여 체적소를 구분하여 표시함은 상기 3차원 객체 데이터를 나타내는 트리의 노드가 부(sub)노드로 구성될 때는 'S' 로 표시하고, 노드내의 부체적소가 모두 객체가 없는 체적소로 구성될 때는 'W'로 표시하고, 노드 내의 부체적소가 모두 객체가 있는 체적소로 구성될 때는 'B'로 표시하고, 노드 내의 부체적소 값이 PPM 알고리즘에 의해 부호화될 때는 'P'로 표시함을 특징으로 한다. 상기 노드 부호화단계는 노드가 S 노드인지 P노드인지에 관한 정보를 부호화하는 단계; 및 상기 노드정보가 S 노드이면 S 노드의 상세정보를 부호화하고, P노드이면 P노드의 상세정보를 부호화하는 단계를 포함함을 특징으로 한다.

<35> 상술한 기술적 과제를 달성하기 위한 본 발명에 의한, 3차원 객체 데이터 부호화 방법은, 포인트텍스처, 체적소, 팔진트리구조의 데이터 중 어느 하나로 이루어진 3차원 객체 데이터에 대해 노드마다 라벨이 부착된 트리구조의 3차원 객체 데이터를 생성하는 트리구조생성단계; 상기 노드마다 라벨이 부여된 트리구조의 3차원 객체 데이터를 상기 라벨을 참조하여

병합하는 단계; 상기 병합된 노드를 부호화하는 노드부호화단계; 상기 노드 부호화된 데이터를 비트스트림으로 생성하는 비트스트림 생성단계; 및 상기 병합 단계, 노드 부호화단계 및 비트스트림생성단계를 상기 트리구조의 3차원 객체 데이터의 최상위 노드까지 반복하는 단계를 포함함을 특징으로 한다. 상기 라벨이 부여된 트리구조의 3차원 객체 데이터를 생성은 바운딩 볼륨안에 객체가 있을 때와 없을 때를 구별하여 표시하되, 트리의 노드가 하위노드를 가질 때는 'S' 로 표시하고, 노드내의 체적소가 모두 객체가 없는 체적소로 구성될 때는 'W'로 표시하고, 노드 내의 체적소가 모두 객체가 있는 체적소로 구성될 때는 'B'로 표시하고, 노드 내의 체적소 값이 PPM 알고리즘에 의해 부호화될 때는 'P'로 표시함을 특징으로 한다. 상기 병합 단계는 노드의 라벨이 'P'인 노드와, 하위 노드의 라벨이 'W'와 'B'인 'S'노드를 병합할 후보노드로 선택하는 단계; 상기 후보노드에 대해 병합 전후의 왜곡된 비트수의 차(??D)와 병합 전후의 비트수의 차(??R)의 비가 최소가 되는 노드를 최적 노드로 선택하는 단계; 상기 최적노드의 라벨을 'B'로 표시하는 단계; 상기 후보노드에서 최적 노드로 선택된 노드를 제외하고 상기 후보노드를 갱신하는 단계를 포함함을 특징으로 한다. 상기 병합 전후의 왜곡된 비트수의 차( $\Delta D$ )는 원래 모델 V와 왜곡측정으로서의 그것의 근사치 간의 해밍 거리(Hamming Distance)를 이용하여 계산하며,

&lt;36&gt;

$$D = \sum_{x=1}^X \sum_{y=1}^Y \sum_{z=1}^Z |V(x,y,z) - \hat{V}(x,y,z)|$$

<37> (여기서  $X \times Y \times Z$  는 모델의 해상도이다.) 에 의해 계산됨을 특징으로 한다.

<38> 상기 노드 부호화는 상기 후보노드가 큐에 존재하는지 여부를 나타내는 정보(Continue 플래그)를 부호화하는 단계; 상기 후보노드를 저장하는 큐 내에서의 위치 정보를 나타내는 노

드 위치정보를 부호화하는 단계; 노드가 S 노드인지 P노드인지에 관한 노드종류정보를 부호화하는 단계; 상기 노드종류정보가 S 노드이면 S 노드의 상세정보를 부호화하고, P노드이면 P노드의 상세정보를 부호화하는 단계를 포함함을 특징으로 한다. 상기 S 노드 부호화는 색상정보의 평균값을 부호화하는 단계; 및

<39> 8개의 부(sub)노드의 라벨을 부호화하는 단계를 포함함을 특징으로 한다. 상기 P 노드 부호화는 깊이정보를 PPM 부호화하는 단계; 및 색상정보를 부호화하는 단계를 포함함을 특징으로 한다. 상기 깊이정보의 PPM 부호화는 소정 개수의 콘텍스트를 사용하여 해당 노드 이하의 전체 하위노드를 래스터 스캔순서로 PPM 부호화함을 특징으로 한다. 상기 색상정보 부호화는 현재 노드의 'B' 체적소들의 색상인 R(Red), G(Green), B(Blue) 값을 DPCM 부호화하고, AAC(Adaptive Arithmetic Coding) 부호화함을 특징으로 한다.

<40> 상술한 기술적 과제를 달성하기 위한 본 발명에 의한, 3차원 객체 데이터 부호화 장치는, 포인트텍스처, 체적소, 팔진트리구조의 데이터 중 어느 하나로 이루어진 3차원 객체 데이터에 대해 노드마다 노드의 종류를 나타내는 라벨이 부착된 트리구조의 3차원 객체 데이터를 생성하는 트리구조 생성부; 상기 트리구조의 3차원 객체 데이터에 대해 상기 라벨을 참조하여 병합하는 병합순서선택부; 상기 병합된 노드를 부호화하는 노드부호화부; 및 상기 노드 부호화된 데이터를 비트스트림으로 생성하는 비트스트림생성부를 포함함을 특징으로 한다. 상기 병합순서선택부는 노드의 라벨이 'P'인 노드와, 하위 노드의 라벨이 'W'와 'B'인 'S'노드를 병합할 후보노드로 선택하는 후보노드선택부; 상기 후보노드에 대해 병합 전후의 왜곡된 비트수의 차( $\Delta D$ )와 병합 전후의 비트수의 차( $\Delta R$ )의 비가 최소가 되는 노드를 최적 노드로 선택하여 상기 최적노드의 라벨을 'B'로 표시하는 최적노드선택부; 상기 후보노드 중 최적 노드로 선택된 노드를 제외하고 상기 후보노드를 갱신하는 후보노드갱신부를 포함함을 특징으로 한다.

상기 노드 부호화부는 압축된 비트스트림의 끝인지 여부를 나타내는 Continue 플래그를 부호화하는 Continue 플래그 부호화부; 상기 후보노드를 저장하는 후보노드 큐를 구비하고, 상기 큐 내에서의 위치 정보를 나타내는 노드 위치정보를 부호화하는 노드위치부호화부; 노드가 S 노드인지 P노드인지에 관한 노드종류정보를 부호화하는 노드SOP선택부; 상기 노드종류정보가 S 노드이면 S 노드의 상세정보를 부호화하는 S노드부호화부; 및 상기 노드종류정보가 P노드이면 P노드의 상세정보를 부호화하는 P노드부호화부를 포함함을 특징으로 한다.

<41> 상술한 기술적 과제를 달성하기 위한 본 발명에 의한, 3차원 객체 데이터 복호화 방법은, 부호화된 3차원 객체 데이터의 비트스트림에서 Continue 플래그 정보를 읽어 복호화하는 단계; 부호화된 3차원 객체 데이터의 비트스트림에 대해 노드종류 정보를 복호화하는 단계; 상기 노드종류 정보가 S 노드를 나타내면 S 노드를 복호화하고, 상기 노드종류 정보가 PPM 노드를 나타내면 PPM 노드를 복호화하는 단계; 및 노드 복호화된 데이터를 트리구조로 복원하는 단계를 포함함을 특징으로 한다.

<42> 상술한 기술적 과제를 달성하기 위한 본 발명에 의한, 3차원 객체 데이터 복호화 방법은, 부호화된 3차원 객체 데이터의 비트스트림에 대해 노드 복호화하는 단계; 및 노드 복호화된 데이터를 트리구조로 복원하는 단계를 포함함을 특징으로 한다. 상기 노드 복호화단계는 부호화된 3차원 객체 데이터의 비트스트림에서 Continue 플래그를 읽어 이를 복호화하는 단계; 노드위치의 값을 읽어 현재 노드가 큐에서 어느 후보노드인지를 나타내는 위치정보로 복호화하는 단계; 노드 종류 정

보를 복호화하는 단계; 상기 노드종류 정보가 S 노드를 나타내면, S 노드를 복호화하는 단계; 및 상기 노드종류 정보가 PPM 노드를 나타내면, PPM 노드를 복호화하는 단계를 포함함을 특징으로 한다. 상기 S 노드 복호화는 상세 정보의 값으로 하위 8개 노드들에 대한 평균 색상을 복호화하고 하위 8개 노드들을 차례대로 B 또는 W로 복호화 함을 특징으로 한다. 상기 PPM 노드를 복호화는 DIB(Detailed Information Bits) 값으로 현재 노드를 PPM 복호화하고 현재 노드의 Black 체적소들의 색상인 R,G,B 값을 역 AAC를 수행하여 복호화하고 역 DPCM를 수행하여 복호화함을 특징으로 한다.

<43> 상술한 기술적 과제를 달성하기 위한 본 발명에 의한, 3차원 객체 데이터 복호화 장치는, 부호화된 3차원 객체 데이터의 비트스트림을 받아들이는 비트스트림독출부; 상기 비트스트림을 복호화하는 노드복호화부; 및 상기 복호화된 노드를 트리구조로 복원하는 트리구조복원부를 포함함을 특징으로 한다. 상기 노드복호화부는 비트스트림이 계속되는 지를 나타내는 Continue 플래그를 복호화하는 Continue 플래그 복호화부; 부호화된 3차원 객체 데이터의 비트스트림에 대해 노드위치의 값을 읽어 현재 노드가 큐에서 어느 후보노드인지를 나타내는 위치 정보로 복호화하는 노드위치정보 복호화부; 노드 종류 정보를 복호화하는 노드종류선택부; 상기 노드종류 정보가 S 노드를 나타내면, 상세정보의 값으로 하위 8개 노드들에 대한 평균 색상을 복호화하고 하위 8개 노드들을 차례대로 Black 또는 White로 복호화하는 S노드복호화부; 및 상기 노드종류 정보가 PPM 노드를 나타내면, 상세 정보값으로 현재 노드를 PPM 복호화하고 현재 노드의 B 체적소들의 색상인 R,G,B 값에 대해 역 AAC를 수행하여 복호화하고 역 DPCM를 수행하여 복호화하는 P노드복호화부를 포함함을 특징으로 한다.

<44> 그리고 상기 기재된 발명을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공한다.



- <45> 본 출원은 미국상표특허청에 가출원된 2건의 출원을 기초출원으로 하여 우선권을 주장하여 출원된다. 이하에서 본 출원의 우선권주장의 기초가 된 2건의 가출원서에 기재된 발명을 기술한다.
- <46> 미국특허가출원번호 제60/430,976호(출원일: 2002. 12. 5, 발명의 명칭: 깊이 이미지에 기초한 표현에 대한 핵심 실험결과(AFXA8.3))에 기술된 발명은 다음과 같다.
- <47> 1. 서론
- <48> DIBR은 복잡한 기하학적 구조를 가진 객체를 표현하고 렌더링하는 새로운 표현방법 및 렌더링방법이다[1]. 3차원 객체를 다각형 메쉬를 이용하여 표현하는 대신, DIBR은 객체를 눈에 보이는 표면을 감싸는(covering) 기준 이미지(reference image) 세트로 표현한다. 통상의 색깔 정보에 부가하여, 각 참조 이미지는 깊이 맵(depth map)으로 나타내어지며, 상기 깊이 맵은 이미지 평면의 픽셀로부터 객체 표면까지의 거리를 배열(array)로 나타낸 것이다. DIBR은 계산이 그리 복잡하지 않고도 참조 이미지가 객체에 대한 양질(high quality)의 시각화(visualization)를 제공할 수 있다는 것이다. 게다가 DIBR 뷰(view)를 렌더링하는 복잡도는 풍경(scene)의 복잡성에는 관계없이 뷰의 픽셀 수(즉 뷰의 해상도)에만 관련된다.
- <49> DIBR 패밀리의 주된 형식(format)은 심플 텍스처(SimpleTexture), 포인트 텍스처(PointTexture) 및 팔진트리 이미지(OctreeImage)이다. 상기 포인트텍스처는 하나의 카메라 위치에서 바라본 픽셀들의 배열로 객체를 표현한다. 각각의 포인트텍스처 픽셀은 깊이, 색깔 및 몇 개의 다른 속성들에 의해 표현된다. 각 시선(line of sight)을 따라 다수의 픽셀들이 있을 수 있다. 따라서 포인트텍스처 이미지는 보통 다수의 층(multiple layer)으로 구성되며, 엄청난 양의 저장공간을 요구한다. 본 발명에서는 포인트 텍스처 이미지에서 깊이 및 색깔 정보의 압축에 초점이 있다.

<50> Duan 과 Li 가 최초로 포인트텍스처 이미지 압축 방법을 제시했다[2]. 그들의 알고리즘은 JPEG-LS 표준을 이용하여 각 시선을 따라 픽셀의 수를 부호화한다[3]. 그리고 영상을 몇 개의 층으로 분해한다. 그리고 나서 깊이 및 컬러 데이터는 모양-적응 웨이블릿 코덱(shape-adaptive wavelet codec)을 이용하여 계층별로 부호화된다. 그러나 분해를 한 후에는 깊이 및 컬러의 상관성(correlation)은 효과적으로 이용되어 질 수 없다. 왜냐하면, 한 계층 내에 있는 인접하는 픽셀들은 3차원 공간에서 서로 멀리 떨어져 있을 수 있기 때문이다.

<51> 포인트 이미지는 볼륨(volume) 데이터와 유사한 구조를 갖는다. 본 발명에서는 포인트 텍스처 이미지는 정보의 손실없이 볼륨데이터로 변환된다. 그리고 볼륨데이터의 정규성을 이용하여 깊이 및 컬러 데이터를 효율적으로 압축한다. 특히 깊이 데이터는 이진 체적소(voxel) 이미지로 변환되며, 상기 체적소 이미지는 차례 차례 팔진트리에 의해 표현된다. 팔진트리의 각 노드는 PPM(Predictive partial matching) 알고리즘을 이용하여 압축된다[4]. 깊이 데이터가 압축된 후에는 컬러 데이터가 너비 우선 탐색(breadth-first traversal)에 기초하여 배열된다. 그리고 나서 차분 펄스 코딩 변조(DPCM) 알고리즘을 이용하여 부호화된다[4].  $8,192 (=2^{13})$  콘텍스트(context)를 가진다.

<52> 2. 깊이 데이터의 압축

<53> 깊이 정보는 트리 구조의 PPM 방법을 이용하여 압축된다. 먼저, 깊이 데이터는 이진 체적소 이미지로 변환된다. 그리고 나서 체적소 이미지는 팔진트리로 표현된다. 각 노드의 팔진트리는 PPM을 이용하여 압축된다. 팔진트리의 깊이는 코딩 이득을 최대화시키기 위해 결정된다.

<54> 2.1 이진 체적소 이미지로의 변환

<55>        깊이를 데이터를 이진 체적소 이미지로 변환하기 위해 경계 볼륨(bounding volume)이 생성된다. 경계 볼륨은 입력 포인트텍스처 이미지와 같은 해상도를 갖는다. 예를 들어 포인트텍스처 이미지가  $X \times Y$ 의 공간 해상도와 각 픽셀의 깊이가  $z$ 의 해상도를 가지면,  $X \times Y \times Z$ 의 크기를 갖는 경계 볼륨이 생성된다. 변환과정은 순차적으로 행해진다. 먼저, 경계 볼륨 내에 있는 모든 체적소 값은 '0'으로 초기화된다. 그리고 나서 포인트 텍스처에 있는 각 픽셀에 대해, 상응하는 체적소 값은 '1'로 세팅된다. 본 발명에서, 객체를 표현할 때 '0' 체적소는 'W' 체적소(white voxel)이라 부르고, '1' 체적소는 'B' 체적소(black voxel)이라 부르기로 한다.

## <56>        2.2 팔진트리 세부 분할

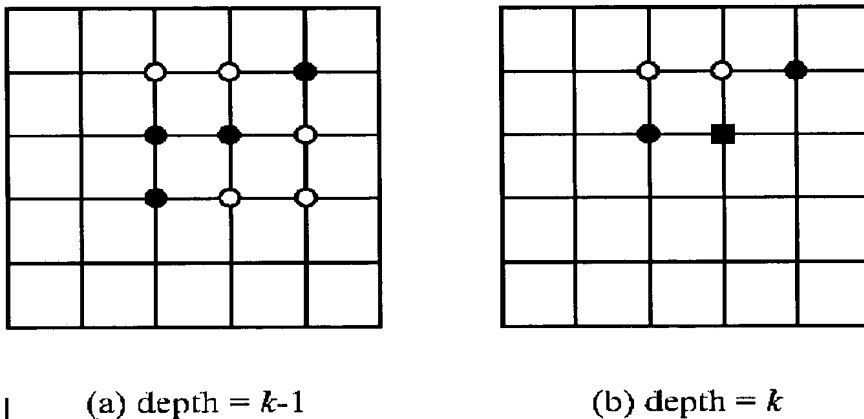
<57>        경계 체적소는 팔진트리를 이용하여 나누어진다. 팔진트리의 루트 노드(root node)는 경계 볼륨에 대응한다. 만일 경계 볼륨이 B 체적소를 포함하고 있다면, 루트 노드는 1로 할당되고, 경계 체적소는 8개의 부(sub)체적소(subvoxel)로 나누어진다. 그렇지 않다면, 루트 노드는 0으로 할당되고 분할 과정은 종료된다. 마찬가지로, 각 서브볼륨이 B 체적소를 포함하고 있다면, 상응하는 노드는 1로 할당되고, 부체적소는 다시 8개의 부체적소로 나누어진다. 아니면, 상응하는 노드는 0으로 할당되며, 부체적소는 나누어지지 않는다. 이러한 과정은 팔진트리가 이미 정해진 깊이까지 확장될 때 까지 계속된다. 그리고 나서 값이 1인 각 말단노드(leaf node)는 PPM알고리즘을 이용하여 부호화된다.

<58>        서브분할은 '0'노드를 잘라내어(pruning) 코딩 이득을 개선하기 위해 사용된다.

## <59>        2.3 PPM 코딩

<60> PPM 코딩에서, 각 체적소 값은 콘텍스트에 기초하여 압축되며, 이전에 부호화된 체적소 값의 수에 의해 정해진다. 도 1은 본 발명에서 사용되고 있는 콘텍스트의 예를 도시한 것이다. 도 1(b)의 사각형 체적소가 부호화된다. 체적소 값을 부호화하기 위해 26개의 인접 체적소 중 이미 부호화된 체적소들이 조건 클래스(conditioning class) 즉 콘텍스트로서 사용된다. 체적소값은 래스터(raster) 스캔 순서로 부호화되기 때문에, 도 1(a)의 9개의 체적소와 도 1(b)의 4개의 체적소가 콘텍스트로 사용된다. 따라서 이 예에서는 콘텍스트가 13 체적소 값('0011101000011')에 의해 주어진다. 이 콘텍스트를 이용하여 사각형 체적소 값이 적응산술 코더(adapitve arithmetic coder)에 의해 압축된다[5].

<61>



<62> 그림 1. 콘텍스트의 예. (b)에서의 사각형 체적소가 부호화되며, (a)와 (b)에 원으로 표시된 13개의 체적소들이 콘텍스트로 사용된다. 속이 찬 검은 원은 1, 속이 하얀 원은 0의 값을 갖는다. 이렇게 부호화하는 방식이 제안하는 TSPPM(Tree Selection PPM) 코딩 방식이다.

<63> 3. 컬러데이터의 압축

<64> 3차원 객체에서, 이웃하는 체적소들의 컬러들은 밀접하게 상관되어 있다. 따라서 상관성을 이용하기 위해서는 FIFO(First-In First-Out) 리스트를 이용하여 체적소들을 스캔하고, 각

체적소의 컬러는 스캔순서로 DPCM 부호화된다. 특히 컬러 데이터는 다음과 같은 방식으로 부호화된다. 초기 체적소가 선택되고, 그 컬러는 고정된 비트수를 사용하여 무손실 부호화된다. 그리고 나서 체적소는 리스트로 넣어진다.(pushed) 체적소는 리스트로부터 선택되며, 그 컬러는 이전 체적소의 컬러로부터 예측되며, 예측 에러는 산술코더(arithmetic coder)로 무손실 부호화된다. 그리고 나서 26개의 인접하는 B 체적소들은 리스트에 일렬로 넣어진다. 이 단계는 리스트가 완전히 빌 때 까지 반복된다.

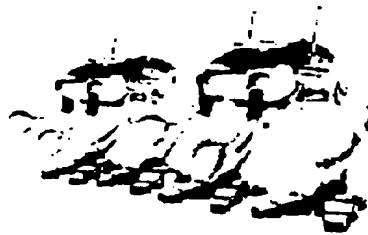
<65> 4. 시뮬레이션 결과

<66> 제안된 알고리즘의 성능은 도 2에 있는 테스트 모델을 사용하여 평가된다. 도 2(a)는 256x256x256 해상도를 가지며, 도 2(b) (d) 는 512x512x512 해상도를 갖는다. 모델의 깊이 해상도는 모두 8비트이다.

<67>

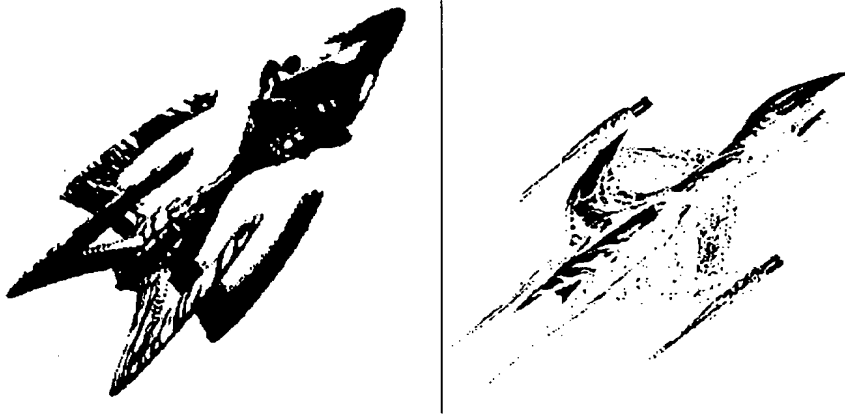


(a) Angel



(b) Robots

&lt;68&gt;



(c) Fighter

(d) WhiteStar

&lt;69&gt; 그림 2. 테스트 모델들

<70> 표 1은 깊이 데이터에 대한 압축성능을 비교하고 있다. 제안된 TSPPM 알고리즘은 범용 사전기반 코덱(universal dictionary-based codec)인 WinZip에 비해 저장공간의 11-14% 만을 요구하고 있음을 알 수 있다. 표 2는 컬러 데이터에 대한 압축성능을 비교하고 있다. 제안된 DPCM 코더는 또한 WinZip 보다 훨씬 작은 비트율을 요구한다.

<71> 표 1. 깊이 이미지에 대한 압축 파일 크기(바이트) PPM = 팔진트리 구조가 없는 순수한 PPM 코딩, TSPPM = 팔진트리 구조를 갖는 PPM, WINZIP = WinZip을 이용한 압축

&lt;72&gt;

	Original	PPM	TSPPM(A)	WINZIP(B)	Ratio(A/B)
Angel	307,733	17,652	17,446	151,029	0.11
Robots	645,243	38,754	37,377	391,866	0.10
Fighter	454,510	18,342	16,778	151,029	0.11
WhiteStar	533,381	31,247	29,534	217,602	0.14

<73> 표2. 컬러 데이터의 압축파일 크기(바이트)

&lt;74&gt;

	Original	DPCM(A)	WINZIP(B)	Ratio(A/B)
Angel	722,046	224,820	676,703	0.33
Robots	1,141,350	642,230	1,594,778	0.40
Fighter	573,030	218,388	782,787	0.27
WhiteStar	806,082	366,001	1,305,585	0.28

&lt;75&gt;

과거의 결과에서는[6], 깊이에 대해서는 압축 결과가 5.2 : 5.4 : 1 이고, 컬러에 대해서는 1.2 : 1 이었다. 표 1과 표 2에 따르면, 원래 데이터로부터 압축비는 깊이에 대해서는 17 - 24 : 1 이고, 컬러에 대해서는 1.8 - 3.2 : 1 이며, 이는 전보다 상당히 높다.

&lt;76&gt;

## 5. 결론

&lt;77&gt;

본 발명에서, 계층화된 깊이 이미지에 대한 무손실 압축 알고리즘을 제안하였으며, 그 각각의 픽셀은 컬러 벡터 뿐만 아니라 깊이 값과 연관된다. 제안된 알고리즘은 깊이 데이터와 컬러 데이터를 분리하여 압축한다. 첫째, 트리 구조의 예측 부분 매칭(Tree-structured predictive partial matching:TSPPM)이라 불리는 알고리즘이 깊이 정보를 효율적으로 부호화하기 위해 개발되었다. 둘째로, 각 픽셀이 3차원 공간에서 근처의 픽셀을 쫓아가는 방법으로 픽셀들이 스캔되었다. 그리고 나서 컬러 데이터는 스캔 순서로 DPCM-부호화된다. 시뮬레이션 결과는 제안된 알고리즘이 범용 코덱인 WinZip 보다 상당히 높은 압축율을 제공한다.

&lt;78&gt;

## 6. 참조문헌

&lt;79&gt;

[1] Y.Bayakovski 등, 정지 및 애니메이션 3차원 객체에 대한 깊이 이미지 기반 표현,Proc. ICIP-2002, Rochester, New York, Sep.2002.

&lt;80&gt;

[2] J.Duan and J.Li 계층화된 깊이 이미지의 압축,Proc. of Data Compression Conference, Snowbird, Utah, Mar.2001,pp.331-340.

- <81> [3] M.Weinberger,G,Seroussi,G.Sapiro,LOCO-I 무손실 압축 알고리즘: JPEG-LS 에서의 원리 및 표준화, to appear in IEEE Trans.Image Processing.
- <82> [4] Khalid Sayood, 데이터압축 소개, 제2판, Academic Press, 2000.
- <83> [5] P.G. Howard and J.S.Vitter, 데이터 압축을 위한 산술코딩, Proc.IEEE,vol.82,no.6,pp.857-865,June 1994.
- <84> [6] 한만진, Alexander Zhirkov, Leonid Levkovich-Maslyuk, 박인규, 깊이 이미지 기반 표현(AFX A8.3)에 관한 코어 실험 결과, ISO/IEC JTC1/SC29/WG11 M8259, Fairfax,USA,May 2002.
- <85> 미국특허 가출원 번호 제60/487,263호(출원일: 2003. 7. 16, 발명의 명칭: 깊이 이미지에 기초한 표현에 있어서의 포인트 텍스처의 압축)에 기술된 발명은 다음과 같다.
- <86> DIBR은 복잡한 기하학적 구조를 가진 객체를 표현하고 렌더링하는 새로운 표현방법 및 렌더링방법이다.[6] DIBR 은 MPEG-4 AFX에 채택되어 왔다.[1] 컴퓨터 그래픽에서는 전형적으로 다각형 메쉬로 객체(object)를 표현하여 왔지만, DIBR 은 3차원 객체를 눈에 보이는 객체의 표면을 감싸는(covering) 참조 이미지(reference image)의 세트로 나타낸다. 각 참조 이미지는 상응하는 깊이 맵(depth map)으로 나타내어지며, 상기 깊이 맵은 이미지 평면의 픽셀로부터 객체 표면까지의 거리를 배열(array)로 나타낸 것이다. DIBR의 잇점 중 하나는 다각형모델을 직접 사용하지 않고도 참조 이미지가 객체에 대한 양질(high quality)의 시각화(visualization)를 제공할 수 있다는 것이다. 게다가, DIBR 뷰(view)를 렌더링하는 복잡도는 풍경(scene)의 복잡성에는 관계없이 뷰의 픽셀 수(즉 뷰의 해상도)에만 관련된다. DIBR 패밀리의 주된 형식(format)은 심플 텍스처(SimpleTexture), 포인트 텍스처(PointTexture) 및 팔진



트리 이미지 (OctreeImage) 이다. 상기 포인트텍스처는 하나의 카메라 위치에서 바라본 픽셀들의 배열로 객체를 표현한다. 각각의 포인트텍스처 픽셀은 색깔, 깊이(픽셀로부터 카메라까지의 거리) 및 포인트텍스처 렌더링을 보조하는 몇 개의 다른 속성들에 의해 표현된다. 각 시선(line of sight)을 따라 다수의 픽셀들이 있을 수 있으며, 따라서 포인트텍스처는 보통 다수의 층(multiple layer)으로 구성된다. 1차원에서의 포인트텍스처의 간단한 예가 도 1에 도시되어 있다. 포인트텍스처는 전형적으로 대량의 데이터를 포함한다. 실제 이미지(realistic image)는 보다 높은 샘플링 밀도와 엄청난 양의 데이터를 요구한다. 그러므로 포인트텍스처 이미지의 압축은 효율적으로 수행되어야 한다. Duan 과 Li 가 포인트텍스처 영상을 압축하는 방법을 제시했는데, 상기 방법은 계층화된 깊이 이미지(Layered Depth Image:LDI)라 불린다.[7] 깊이 정보를 압축하기 위해, JPEG-LS 알고리즘이 적용되었다.[4] 또한 색깔(color)정보는 현존하는 코딩(coding) 표준(MPEG-4 SA-DCT[2]와 JPEG-2000[3])을 사용하여 압축된다. 그러나 이러한 알고리즘은 점진적(progressive) 압축 및 전송을 지원하지 못한다.

<87> 대부분의 3차원 객체들은 인터넷이나 무선 채널을 통해 전송된다. 따라서 점진적 코딩 스킴을 개발하는 것이 바람직하다. 상기 점진적 코딩 스킴은 적응(adaptive) 전송 비트율과 대화식의(interactive) 3차원 모델 브라우징을 용이하게 한다. 본 발명에서 포인트 텍스처 이미지의 점진적 압축에 대한 트리 구조의 예측 부분 매칭(TSPPM)을 개발하였다. PPM은 무손실 방식으로 팩시밀리 데이터를 전송하기 위해 원래 Cleary와 Witten 에 의해 제안되었다.[5] PPM을 트리 구조의 코딩과 결합하여, 제안된 알고리즘은 3차원 깊이 정보를 점진적으로 단일의 비트스트림으로 압축한다. 또한 제안된 알고리즘은 차분 펄스 코딩 변조(DPCM)를 사용하여 컬러정보를 압축하고[5], 깊이 정보와 컬러 정보를 섞는다.(interweave) 시뮬레이션 결과는 본 발명에 따른 알고리즘이 유망한 압축성능을 제공함을 보여준다.

<88> 2. 팔진트리 생성

<89> 본 발명은 포인트 텍스처 이미지가 먼저 볼륨데이터로 변환되고, 이것은 팔진트리에 의해 번갈아 가며 표현된다. 그리고 나서 팔진트리의 말단 노드(leaf node)가 PPM 방법에 의해 효율적으로 부호화된다.

<90> 2.1 볼륨데이터로의 변환

<91> 3차원 객체의 깊이정보를 볼륨데이터로 변환하기 위해, 경계 볼륨(bounding volume)이 만들어진다. 경계볼륨은 포인트텍스처와 같은 해상도를 가진다. 예를 들어 만일 포인트텍스처가  $X \times Y$  해상도의 영상을 갖고, 각 픽셀의 깊이정보의 해상도가  $Z$ 이라면,  $X \times Y \times Z$  경계 볼륨이 생성된다. 경계 박스의 원점은 하단왼쪽정면코너(lower-left-front corner)에 위치한다. 오른쪽 체적소는 왼쪽 체적소보다 큰  $x$  값을 가지고, 상단 체적소는 하단 체적소보다 큰  $y$  값을 가지고, 배면(rear)체적소는 정면체적소보다 큰  $z$  값을 가진다. 깊이정보를 이진 볼륨데이터로의 변환은 직접 할 수 있다. 볼륨데이터의 모든 체적소는 'W'('0')으로 초기화된다. 그리고 나서 체적소의 위치가 포인트 텍스처에 있는 픽셀에 의해 참조된다면 체적소는 'B'('1')로 설정된다. 그러므로 'B' 체적소는 3차원 객체에서 점들을 나타내며, 'W' 체적소는 투명한 배경을 나타낸다.

<92> 2.2 초기 트리(Initial Tree)

<93> 볼륨데이터는 팔진트리에 의해 표현되며, 팔진트리의 각 노드는 표1의 4개의 카테고리 중 하나로 분류된다. 첫째로, 만일 경계 볼륨이 객체를 포함하고 있다면, 루트 노드는 'S'로 표시되며, 볼륨은 8개의 동일한 크기의 볼륨으로 세분된다. 만일 세분된 볼륨이 단지 'B' 체적소 또는 'W' 체적소만 포함하고 있으면, 상응하는 노드는 'B' 또는 'W'로 표시된다. 그렇지

않으면, 그 노드는 'S'로 설정되며, 볼륨은 8개의 더 작은 볼륨으로 다시 세분된다. 이 과정은 트리가 미리 설정된 깊이로 갈 때 까지 반복된다. 이러한 깊이에서, 만일 노드가 모두 'B'와 'W' 체적소들을 포함하고 있다면, 'P'로 표시되며, 그 체적소 값은 PPM방법에 의해 부호화될 수 있다. PPM방법은 3.2절에서 설명한다.

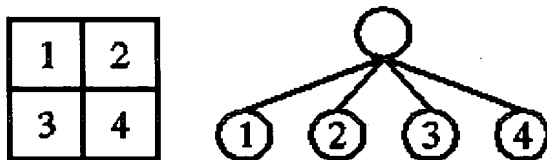
<94> 표 1. 팔진트리 노드의 4가지 라벨

<95>

라벨	설 명
S	Split : 노드가 8개의 서브 노드로 나누어진다.
W	White : 노드가 모두 white <u>체적소들로</u> 구성된다.
B	Fill Black : 노드가 모두 또는 대략 black <u>체적소들로</u> 구성된다
P	PPM : 노드 내의 <u>체적소 값</u> 이 PPM 알고리즘에 의해 부호화된다.

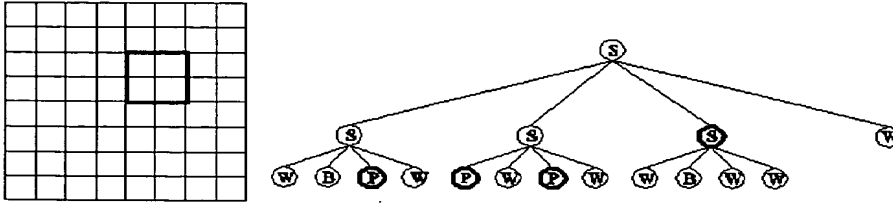
<96> 그림 1은 볼륨데이터의 팔진트리 표현을 나타내고 있다. 간략화를 위해, 본 예에서는 2차원 이진 이미지와 이것들에 대한 사진트리(quadtrees) 표현을 사용한다. 그림 1(a)는 사진트리에서의 부모-자식(parent-children) 관계를 보여주고 있다. 그림 1(b)는 이진 이미지와 상응하는 사진트리를 나타내며, 미리 설정된 깊이는 2이다. 이진 이미지는 사진트리와 세 개의 'P' 노드에 대한 PPM 부호화된 데이터를 사용하여 무손실로 복원될 수 있다.

<97>



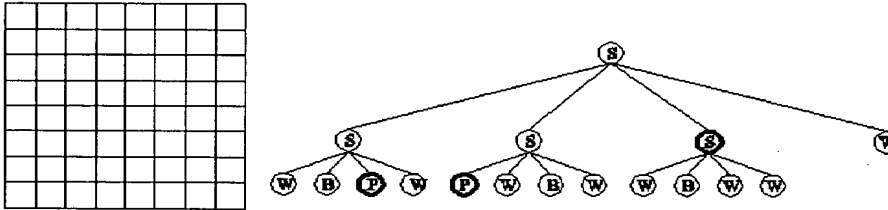
<98> (a) 부모-자식(parent-child) 관계

&lt;99&gt;



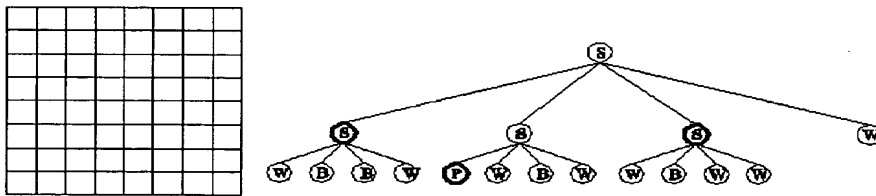
&lt;100&gt; (b) 원래 모델 및 그에 상응하는 트리

&lt;101&gt;



&lt;102&gt; (c) 근사화된 모델 및 병합된(merged) 트리

&lt;103&gt;



&lt;104&gt; (d) 근사화된 모델 및 병합된(merged) 트리

&lt;105&gt; 그림 1 블록 데이터의 팔진트리 표현

&lt;106&gt; 2.3 트리병합(Tree Merging)

<107> 최초의 트리는 원래의 깊이 정보를 몇 개의 정제(refinement) 레벨에서 근사화시키기 위해 반복적으로 병합된다. 병합가능한 노드는 'P'로 표시되는 노드로 정의되거나 그것의 모든 자식들은 'B' 또는 'W' 노드이다. 예를 들어, 도 1(b)에, 트리에 푸른 원으로 표시된 4개의 병합가능한 노드들이 있다. 그들 중 하나의 노드 즉 이진 이미지에서 적색 사각형에 의해 표시된 노드가 'B' 노드로 병합된다고 가정해보자. 결과로 나타난 이미지와 나무는 도 1(c)에 나타나 있다. 그리고 나면 세 개의 병합가능한 노드가 남고, 그들 중 하나는 도 1(d)와 같이 병합된다. 상단좌측사분면에 대한 노드는 이 단계에서 병합되며, 그것의 모든 자식들은 'B' 또는 'W' 노드이다. 병합과정은 모든 노드가 하나의 루트 노드로 병합될 때까지 계속된다.

<108> 각 단계에서, 병합되고 있는 노드는 병합가능한 노드들의 세트에서 선택되며, 다음과 같은 비율로 최소화된다.

$$\text{<109> } -\frac{\Delta D}{\Delta R} = -\frac{D_a - D_b}{R_a - R_b} \quad (1)$$

<110> 여기서  $\Delta R = R_a - R_b$  은 병합후의 비트수( $R_a$ )과 병합전의 비트수( $R_b$ ) 간의 차를 나타낸다. 마찬가지로  $\Delta D = D_a - D_b$  은 병합전후의 왜곡의 차를 나타낸다. 노드의 병합은 요구되는 비트rate는 감소시키지만 근사화된 모델의 왜곡은 증가시킨다. 그러므로 왜곡의 증가는 최소화시키고, rate 감소는 최대화시키는 병합노드를 찾아낸다.

<111>  $\Delta D$ 를 계산하기 위해, 원래 모델  $V$ 와 왜곡측정으로서의 그것의 근사치 간의 해밍 거리를 사용한다.

$$\text{<112> } D = \sum_{x=1}^X \sum_{y=1}^Y \sum_{z=1}^Z |V(x, y, z) - \hat{V}(x, y, z)|$$

<113> 여기서  $X \times Y \times Z$  는 모델의 해상도이다.

<114>  $\Delta R$ 을 계산하기 위해, 팔진트리와 그것의 PPM노드를 표현하기 위한 요구되는 비트수를 계산하는 방법이 필요하다. 그러나 비트율의 정확한 계산은 너무 복잡하다. 그러므로 전형적인 트리비트와 PPM비트가 몇 개의 테스트 모델을 사용하여 계산된다.

<115> 3. 부호화 알고리즘

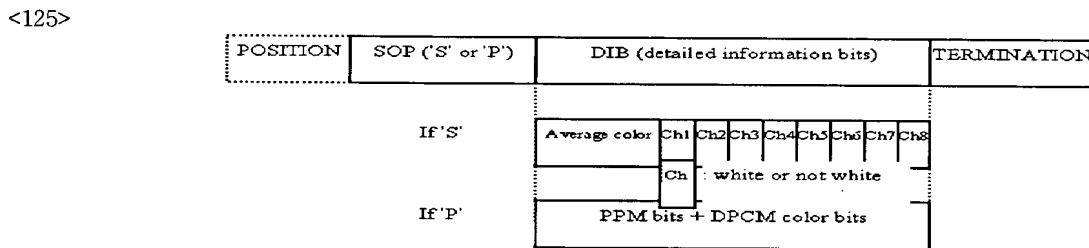
<116> 병합과정에서, 노드는 수학적 1에서의 우선권(priority)에 따라 병합된다. 비트스트림은 역방향으로 부호화된다. 즉, 가장 늦게 병합된 것이 가장 먼저 부호화된다. 따라서 압축된 비트스트림을 구문분석(parsing)하면, 디코더는 가장 거친 모델로부터 가장 정교한 모델(원래의 모델)까지 3차원 모델을 복원할 수 있다.

<117> 3.1 비트스트림의 구성

<118> 압축된 비트스트림은 팔진트리의 노드데이터로 구성된다. 비트스트림의 노드의 순서는 상술한 병합되는 노드의 역순으로 주어진다. 도 2에 도시된 바와 같이 노드의 정보는 POSITION, SOP('S' 또는 'P'), DIB(detailed information bits) 및 TERMINATION 데이터로 구성된다.

<119> 먼저, POSITION 데이터는 B 노드가 분리(split) 또는 PPM 부호화되어야 함을 나타낸다. 예를 들어 디코더가 지금까지 수신된 도 1(d)에 도시된 근사화된 모델을 복원한다고 가정하자. 그리고 나서, 팔진트리에 네 개의 B 노드가 있으며, 인코더가 도 1(c)에 있는 보다 정교한 모델을 얻기 위해 디코더에게 두번째 B 노드가 PPM 부호화되어야 함을 알려주어야 한다. 따라서 B 노드 후보 중 현재의 노드 순서는 POSITION 데이터로 부호화된다. 만일 한 개의 부호 B 노드만 있다면, 위치데이터는 필요없으며, 따라서 부호화되지 않는다.

- <120> SOP는 현재의 노드가 8개의 자식들 노드로 나누어져야하는지 PPM-부호화되어야 하는지를 나타낸다.
- <121> 만일 SOP가 현재노드가 'S'노드임을 나타내면, DIB는 현재 노드의 평균 컬러와 자식들 노드가 W인지 여부를 나타내는 8개의 플래그로 구성된다. W 노드가 아닌 노드는 일시적으로 B 노드로 간주되며, 다음 비트스트림을 위한 후보 B노드를 기록하고 있는 큐에 넣어진다.
- <122> 만일 SOP가 현재노드가 'P'노드임을 나타내면, 현재 노드안에 있는 B 체적소의 깊이 정보가 PPM 부호화되며, B 체적소의 컬러정보는 다음절에서 설명하는 바와 같이 DPCM 부호화된다.
- <123> 마지막으로, TERMINATION은 이것이 압축된 비트스트림의 끝인지 여부를 나타낸다.
- <124> 이 모든 데이터는 적응 산술 코더(adaptive arithmetic coder)를 사용하여 엔트로피 부호화된다[8].



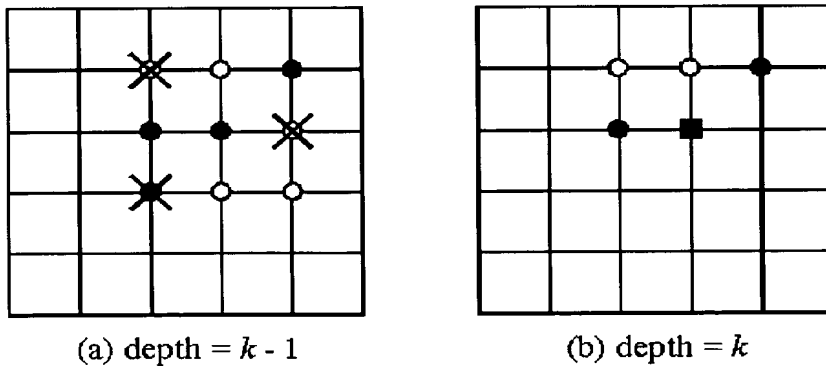
<126> 그림 2. 노드에 대한 데이터 구조

### <127> 3.2 PPM('P')노드의 부호화

- <128> 앞서 언급한 바와 같이, 깊이 정보는 이진 체적소값으로 표현된다. W 체적소는 투명한 배경을 나타내며, B 체적소는 3차원 객체를 나타낸다. 'P' 노드 안에 있는 이진 체적소 값은 이웃하는 체적소 값을 컨텍스트로 이용하여 PPM 부호화된다.

<129> 그림 3은 콘텍스트의 가장 간단한 예 중의 하나이다. 도 3(b)의 사각형 체적소가 부호화될 것이다. 사각형 체적소를 부호화하기 위해, 13개의 원으로 된 체적소가 콘텍스트로 사용된다. 래스터(raster) 스캔순서에 따라 W 체적소는 '0'으로 설정되며, B 체적소는 '1'로 설정된다. 이들 13비트는 사각형 체적소의 콘텍스트로 사용된다. 이 예에서, 사각형 체적소의 콘텍스트는 '0011101000011'이다. 그러나 이 예에서 콘텍스트의 수는  $2^{13}$  이다. 이것은 연산, 처리하기에 너무 크다. 따라서 콘텍스트의 수를 줄일 필요가 있다. 시뮬레이션에 의해, 전체 엔트로피를 크게 증가시키지 않는 체적소의 제거를 선택한다. 먼저, 두 개의 체적소가 제거된다. 그리고 나서 또 다른 체적소가 제거된다. 제거된 세 개의 체적소는 도 3(a)에 도시된 바와 같이 'X'로 마킹된다. 이 콘텍스트를 이용하여 체적소는 콘텍스트 기반 적응 산술코더(AAC)에 의해 압축된다.[8]

<130>



<131> 그림3 콘텍스트의 예 : (b)에 도시된 사각형이 부호화될 것이며, (a)와 (b)에 있는 원은 콘텍스트를 만들기 위해 사용된다. 그리고 (a)는 'X'로 마킹된 세개의 제거된 체적소를 보여준다.



<132> 깊이 정보를 부호화한 후, 현재 노드 안에 있는 B 체적소의 R,G,B 컬러값은 래스터 스캔 순서로 DPCM 부호화된다. 보다 구체적으로 말하면, B 체적소의 R,G,B 값은 이전의 B 체적소 각각의 값들로부터 예측된다. 그리고 예측 오차(residual)는 산술 코더와 함께 부호화된다.

<133> 4. 디코딩 알고리즘

<134> 디코딩 과정은 인코딩 과정과 유사하다.

<135> 먼저, POSITION과 SOP 데이터로부터 디코더는 어떤 B 노드가 분리되어야 하는지 또는 PPM 복호화되어야 하는지를 찾아낸다. 'S'노드의 경우에, 현재 노드의 모든 체적소의 컬러 값은 일시적으로 DIB 내에서 R,G,B 컬러 값의 평균값으로 설정된다. 이 컬러 값들은 자식들 데이터가 수신될 때 갱신된다. 'P'노드의 경우에, 현재 노드의 B 체적소의 깊이 및 컬러 정보는 DIB를 사용하여 손실없이 복원된다. 마지막으로, 만일 TERMINATION이 비트스트림의 끝을 가리킨다면, 디코딩이 종료된다. 그렇지 않다면, 디코더는 다음 B 노드의 디코딩을 계속한다.

<136> 5. 시뮬레이션 결과

<137> 제안된 알고리즘의 성능은 그림 4에 도시된 테스트 모델을 사용하여 평가된다. 그림 4(a)-(e)에 도시된 모델들은 해상도 256x256x256을 가지며, 그림 4(f) (h)는 512x512x512 해상도를 가진다.

<138> 표 2는 포인트텍스처 이미지 상에서의 WinZip과 제안된 알고리즘의 압축성능을 비교하고 있다. 깊이정보에 대해, 제안된 알고리즘은 WinZip 의 파일 크기의 22-59 % 만 요구한다. 컬러 정보에 대해, 제안된 알고리즘은 또한 WinZip 의 파일 크기의 59-81 % 만 요구한다. 유일한 예외는 'Plane' 모델이다. 이것은 다양하지 않은 단지 몇 개의 색깔만 포함하며, 따라서

사전기반 코더인 WinZip 이 더 나은 성능을 나타낸다. 'Plane' 모델을 제외하고, 제안된 알고리즘은 깊이와 컬러 정보를 모두 압축하기 위해 WinZip 의 파일 크기의 49-75 % 만 요구한다.

<139> 제안된 알고리즘은 WinZip이 제공하지 못하는 점진적 전송과 디코딩을 지원한다는 점에서 의미 있다. 그림 5는 'Angel' 모델의 점진적 디코딩을 보여주고 있다. 데이터비트가 수신됨에 따라 복잡하면서 관심영역인 얼굴이 다른 부드러운 부분보다 먼저 정교하게 나타나고 있음을 볼 수 있다. 점진성은 바람직하다. 왜냐하면 그것은 대역폭이 제한된 환경에서 비트스트림의 적응성(Adaptation)과 3차원 모델의 브라우징을 용이하게 해주기 때문이다.

<140> 6. 결론

<141> 포인트 텍스처를 위한 점진적 압축 알고리즘을 제안했다. 그 각각의 픽셀은 컬러 벡터 뿐만 아니라 깊이 값과 연관되어 있다. 제안된 알고리즘은 PPM 방법을 팔진트리 구조로 통합 하므로써 깊이 및 컬러 데이터 둘 다를 효율적으로 부호화한다. 시뮬레이션 결과는 제안된 알고리즘이 범용 Lempel-Ziv 코더인 WinZip 보다 훨씬 높은 압축율을 제공함을 보여주고 있다.

<142> 7. 참조문헌

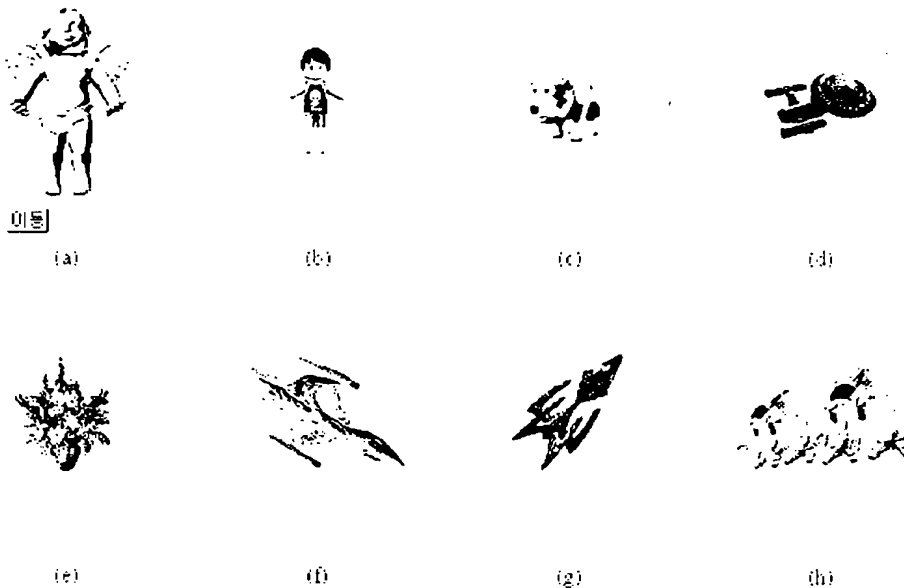
<143> [1] ISO/IEC JTC1/SC29/WG11 N5071 : ISO/IEC 14496-1 : 2001/FPDAM4, Klagenfurt, July 2002.

<144> [2] MPEG-4 Video, MPEG-4 비디오 검증 모델 버전 13.0, ISO/IEC JTC1/SC29/WG11 N2687, 1999년 3월.

<145> [3] JPEG VM ad-hoc, JPEG 2000 검증 모델 5.2 (technical description), ISO/IEC JTC1/SC29/WGIN1422, Maui, Hawaii, 1999년 12월

- <146> [4] M.Weinberger.G. Seroussi, G.Sapiro, LOCO-I 무손실 이미지 압축 알고리즘:  
JPEG-LS 로의 원리 및 표준화, to appear in IEEE Trans. Image Processing.
- <147> [5] K.Sayood, Introduction to Data Compression 제2판, Academic Press, 2000
- <148> [6] Y.Bayakovski 등,정지 및 애니메이션 3차원 객체에 대한 깊이 이미지 기반 표현,  
Proc ICIP-2002, Rochester, 뉴욕, 2002년 9월
- <149> [7] J.Duan J.Li 계층화된 깊이 이미지의 압축, Proc. IEEE, vol.82, no.6,pp.857-865,  
June 1994
- <150> [8] P.G. Howard and J.S.Vitter, 데이터 압축을 위한 산술 코딩, Proc. IEEE, vol.82,  
no.6, pp.857-865, 1994년 6월.

<151>



<152> 그림4. 테스트 모델들

<153> 표 2. 압축된 파일 크기의 비교. P = 손실이 있는 제안된 알고리즘, W = WinZip, R = ratio = P/W. 파일 크기는 바이트

<154>

	DEPTH			COLOR			TOTAL		
	P	W	R	P	W	R	P	W	R
Angel	34,893	150,853	0.23	238,748	402,118	0.59	273,641	552,971	0.49
Avatar	6,396	14,407	0.44	35,813	46,475	0.77	42,209	60,882	0.69
Dog	11,301	34,256	0.34	64,312	112,734	0.57	75,813	146,990	0.52
Plane	5,254	9,771	0.54	51,256	47,245	1.08	56,510	57,016	0.99
Flower	17,804	30,243	0.59	65,016	80,721	0.81	82,820	110,964	0.75
WhiteStar	75,937	250,174	0.30	441,442	725,630	0.61	517,379	975,804	0.53
Fighter	38,108	173,124	0.22	275,853	414,341	0.67	313,961	587,365	0.53
Robots	105,438	330,815	0.32	669,236	939,136	0.71	774,694	1,262,951	0.61

<155>



(a) 50%



(b) 60%



(c) 70%



80%



90%



100%

&lt;156&gt;

## 그림 5 재구성된 Angel 이미지

&lt;157&gt;

이하에서, 본 발명에 따른 포인트텍스처, 체적소, 트리구조의 데이터 중 어느 하나로 이루어진 3차원 객체 데이터 부호화 및 복호화 방법 및 장치에 대해 보다 상세하게 설명한다.

&lt;158&gt;

도 11은 본 발명에 의한 3차원객체 데이터 부호화 장치를 블록도로 도시한 것으로서, 트리구조 생성부(Tree structure generator, 1100), 병합순서선택부(merging order selector, 1110), 노드부호화부(node encoder, 1120) 및 비트스트림 생성부(bitstream generator, 1130)를 포함하여 이루어진다.

&lt;159&gt;

상기 트리구조 생성부(1100)는 3차원 객체 데이터를 표현하고 있는 포인트텍스처, 체적소 및 트리구조 데이터 중 어느 하나를 입력으로 하여 노드마다 노드의 종류를 나타내는 라벨(label)이 부착된 트리구조의 3차원 객체 데이터를 생성한다. 본 실시예에서는 상기 트리구조는 팔진트리구조를 갖는다.

&lt;160&gt;

먼저, 팔진트리구조의 생성을 설명한다. 포인트텍스처 이미지는 먼저 볼륨 데이터로 변환된다. 상기 볼륨 데이터는 교대로 팔진트리로 표현된다. 그리고 나서 팔진트리의 말단 노드(leaf node)들이 PPM 방식에 의해 효율적으로 부호화된다. 그러므로 본 발명에 의한 3차원 객체의 부호화 및 복호화 방법은 포인트텍스처 모델과 팔진트리 모델을 압축하는 것과 동일하다.

&lt;161&gt;

3차원 객체의 깊이정보를 볼륨데이터로 변환하기 위해, 바운딩 볼륨(bounding volume)이 만들어진다. 바운딩 볼륨은 포인트텍스처와 같은 해상도를 가진다. 예를 들어 만일 포인트텍스처가  $X \times Y$  해상도의 이미지를 가지고, 각 픽셀의 깊이정보의 해상도가  $Z$  라면,  $X \times Y \times Z$  바운딩

볼륨이 생성된다. 바운딩 박스의 원점은 정면의 좌측 하단 코너(lower-left-front coner)에 위치한다. 오른쪽 체적소는 왼쪽 체적소보다 큰 x 값을 가지고, 상단 체적소는 하단 체적소보다 큰 y값을 가지고, 배면(rear) 체적소는 정면 체적소보다 큰 z값을 가진다. 깊이정보를 이진 볼륨데이터로의 변환하는 것은 바로 할 수 있다. 볼륨데이터의 모든 체적소는 W(white,'0')로 초기화된다. 그리고 나서 체적소의 위치가 포인트 텍스처에 있는 픽셀에 의해 참조된다면 체적소는 B(black,'1')로 설정된다. 그러므로 B 체적소는 3차원 객체에서 점들을 나타내며, W 체적소는 투명한 배경을 나타낸다.

<162> 볼륨데이터는 팔진트리에 의해 표현되며, 표1은 팔진트리 노드의 4가지 라벨을 나타내고 있다. 팔진트리의 각 노드는 표 1의 4개의 카테고리 중 하나로 분류된다. 첫째로 만일 바운딩 볼륨이 객체를 포함하고 있다면 루트 노드는 'S'로 표시되며, 볼륨은 8개의 동일한 크기의 볼륨으로 세분된다. 만일 세분된 볼륨이 단지 'B' 체적소 또는 'W' 체적소만 포함하고 있으면, 상응하는 노드는 'B' 또는 'W'로 표시된다. 그렇지 않으면 그 노드는 'S'로 설정되며, 볼륨은 8개의 더 작은 볼륨으로 다시 세분된다. 이 과정은 트리가 미리 설정된 깊이로 갈 때까지 반복된다. 이러한 깊이에서 만일 노드가 모두 'B'와 'W' 체적소들을 포함하고 있다면 'P'로 표시하고, 그 체적소 값은 PPM방법에 의해 부호화할 수 있다.

<163> 【표 1】

라벨	설 명
S	Split : 노드가 8개의 서브 노드로 나누어진다.
W	White : 노드가 모두 white 체적소들로 구성된다.
B	Fill Black : 노드가 모두 또는 대략 black 체적소들로 구성된다
P	PPM : 노드 내의 체적소 값이 PPM 알고리즘에 의해 부호화된다.

- <164> 도 3은 라벨이 부착된 블록데이터의 팔진트리 표현을 나타내고 있다. 설명을 용이하게 하기 위해, 본 예에서는 2차원 이진 이미지와 이것들에 대한 사진트리(quadtrees) 표현을 사용한다. 도 3(a)는 사진트리에서의 부모-자식(parent-children) 관계를 보여주고 있다. 도 3(b)는 이진 이미지와 상응하는 사진트리를 나타내며, 미리 설정된 깊이는 2이다. 이진 이미지는 사진트리와 세 개의 'P'노드에 대한 PPM 부호화된 데이터를 사용하여 무손실로 복원될 수 있다.
- <165> 도 12는 상기 병합순서선택부(1110)의 구성을 블록도로 도시한 것으로서, 상기 병합순서선택부(1110)는 트리구조의 3차원 객체 데이터에 대해 상기 라벨을 참조하여 병합(merging)하며, 후보노드 선택부(candidate node selector, 1200), 최적노드 선택부(optimal node selector, 1210) 및 후보노드 갱신부(candidate node updater, 1220)로 이루어진다. 상기 후보노드 선택부(1200)는 노드의 라벨이 'P'인 노드와 'S' 노드를 병합할 후보노드로 선택한다. 단 상기 'S'노드는 하위(sub) 노드의 라벨이 'W'또는 'B' 또는 W'와 'B'로 구성된다. 그리고 상기 최적노드선택부(1210)는 상기 후보노드에 대해 병합을 전후하여 왜곡된 비트수의 차(??D)와 병합 전후의 비트수의 차(??R)의 비가 최소가 되는 노드를 최적 노드로 선택하여 상기 최적노드의 라벨을 'B'로 표시한다. 상기 후보노드갱신부(1220)는 상기 후보노드 중 최적 노드로 선택된 노드를 제외하고 상기 후보노드를 갱신한다.
- <166> 도 3과 도 13 및 도 14를 참조하여 트리 병합(Tree Merging)에 대해 설명한다. 도 3은 라벨이 부착된 팔진트리표현과 트리 병합을 도시한 것이다. 최초의 트리는 원래의 깊이 정보를 몇 개의 정제(refinement) 레벨에서 근사화시키기 위해 반복적으로 병합된다. 병합가능한 노드는 'P'로 표시되는 노드이거나 그것의 모든 자식들은 'B'또는 'W'노드인 'S'노드이다. 예를 들어, 도 3(b)에, 트리에 굵은 흑색 원으로 표시된 4개의 병합가능한 노드들이 있다. 그들 중 하나의 노드 즉 이진 이미지에서 굵은 흑색 사각형에 의해 표시된 노드가 'B'노드로 병합된다

고 가정해보자. 결과로 나타난 이미지와 트리는 도 3(c)에 나타나 있다. 그리고 나면 세 개의 병합 가능한 노드가 남고, 그들 중 하나는 도 3(d)와 같이 병합된다. 상단 좌측 사분면에 대한 노드는 이 단계에서 병합되며, 그것의 모든 자식들은 'B' 또는 'W'노드이다. 병합과정은 모든 노드가 하나의 루트 노드로 병합될 때까지 계속된다.

<167> 도 13은 상술한 과정인 후보노드 선택부(1200)의 동작을 흐름도로 도시한 것이다. i는 노드의 번호를 나타내고 count는 후보노드의 수를 나타낸다. i 값과 count 값을 0으로 초기화하고(1300단계), 노드가 'S'이거나 'P'이면(1310단계), 후보노드로 선택되어 큐에 저장되고 count 값이 하나 증가한다.(1330단계) 여기서 'S'노드는 하위노드가 'W' 또는 'B'로 또는 이들이 섞여서 있는 경우만 후보노드가 될 수 있다. 만일 노드가 'S'이거나 'P'가 아니면, 다음 노드에 대해 상기 1310단계를 수행한다.(1320단계) 상기의 과정을 마지막 노드까지 반복한다.(1340단계)

<168> 한편 각 단계에서 병합되고 있는 노드는 병합가능한 노드들의 세트에서 선택되며 다음과 같은 비율로 최소화된다.

<169>

$$-\frac{\Delta D}{\Delta R} = -\frac{D_a - D_b}{R_a - R_b}$$

【수학식 1】

<170> 여기서  $\Delta R = R_a - R_b$  는 병합후의 비트수( $R_a$ )과 병합전의 비트수( $R_b$ ) 간의 차를 나타낸다. 마찬가지로  $\Delta D = D_a - D_b$  은 병합전후의 왜곡의 차를 나타낸다. 노드의 병합은 요구되는 비율(rate)은 감소시키지만 근사화된 모델의 왜곡은 증가시킨다. 그러므로 왜곡의 증가는 최소화시키고, rate 감소는 최대화시키는 병합노드를 찾아낸다.



<171>  $\Delta D$ 를 계산하기 위해 원래 모델  $V$ 와 왜곡측정으로서의 그것의 근사치 간의 해밍 거리를 사용한다.

<172>

$$D = \sum_{x=1}^X \sum_{y=1}^Y \sum_{z=1}^Z |V(x,y,z) - \hat{V}(x,y,z)|$$

【수학식 2】

<173> 여기서  $XxYyZz$  는 모델의 해상도이다.

<174>  $\Delta R$ 을 계산하기 위해 팔진트리와 그것의 PPM노드를 표현하기 위한 요구되는 비트수를 계산하는 방법이 필요하다. 그러나 비트율의 정확한 계산은 너무 복잡하다. 그러므로 전형적인 트리비트와 PPM 비트가 몇 개의 테스트 모델을 사용하여 계산된다.

<175> 도 14는 병합순서선택부(1210)의 동작을 흐름도로 도시한 것이다. 후보노드선택부(1200)에서 후보노드가 선택되면 후보노드 각각에 대해 수학식 1에 의해 계산된 값이 최소가 된 후보노드를 최적노드로 선택하여(1410단계), 선택된 노드를 'B'로 하고(1420단계), 상기 노드는 큐에서 제거한 후(1430단계), 큐를 갱신한다(1440단계). 상기 과정을 모든 노드가 병합될 때까지 계속한다. (1450단계)

<176> 한편, 상기 노드부호화부(1120)는 병합된 노드를 부호화한다. 도 15는 상기 노드부호화부(1120)의 구성을 도시한 것으로서, Continue 플래그 부호화부(1500), 노드위치부호화부(1510), 노드SOP선택부(1520), S노드부호화부(1530) 및 P노드부호화부(1540)로 이루어진다.

<177> 상기 Continue 플래그 부호화부(1500)는 압축된 비트스트림의 끝인지 여부를 나타내는 Continue 플래그를 부호화한다. 끝이 아니면 참(True)로 설정되고, 끝이면 거짓(false)로 설정된다. 상기 노드위치부호화부(1510)는 상기 후보노드를 저장하는 후보노드 큐를 구비하고, 상기 큐 내에서의 위치 정보를 나타내는 노드 위치정보를 부호화한다. 상기 노드SOP선택부

(1520)는 노드가 S 노드인지 P노드인지에 관한 노드종류정보를 부호화한다. 상기 S노드부호화부(1530)는 상기 노드종류정보가 S 노드이면 S 노드의 상세정보(Detailed Information Bits:DIB)를 부호화한다. 상기 P노드부호화부(1540)는 상기 노드종류정보가 P노드이면 P노드의 상세정보(DIB)를 부호화한다.

<178> 도 16은 상술한 노드 부호화부(1120)의 동작을 흐름도로 도시한 것이다. 이를 간략히 설명하면, 먼저 Continue 플래그를 부호화한다.(1600단계) 상기 Continue 플래그는 압축된 비트스트림의 끝인지 여부를 나타낸다. 끝이 아니면 참(True)을 나타낸다. 그리고 나서 노드위치정보(position)를 부호화한다.(1600단계) 그리고 나서 노드가 'P'노드인가 조사하여(1620단계), P노드이면 P노드 부호화를 하고(1630단계), 그렇지 않으면 S노드 부호화한다.(1640단계) 상기 과정을 모든 노드가 부호화될 때 까지 반복한다.(1650단계)

<179> 그리고 도 17은 S노드부호화부(1520)의 동작을 흐름도로 도시한 것이다. S노드의 부호화는 컬러정보의 평균값을 부호화하고(1700단계), 8개의 하위노드(subnode)의 라벨을 부호화한다.(1710단계) 도 18은 P노드부호화부(1530)의 동작을 흐름도로 도시한 것으로서, 깊이정보를 부호화하고(1800단계), 컬러정보를 DPCM 부호화한다.(1810단계)

<180> 이를 예를 들어 보다 상세히 설명하기로 한다. 병합과정에서, 노드는 수학적 1에서의 우선권(priority)에 따라 병합된다. 비트스트림은 역방향으로 부호화된다. 즉 가장 늦게 병합된 것이 가장 먼저 부호화된다. 따라서 압축된 비트스트림을 구문분석(parsing)하면, 디코더는 가장 거친 모델로부터 가장 정교한 모델(원래의 모델)까지 3차원 모델을 복원할 수 있다. 도 4는 상기 병합 방법을 이용하여, 비트수-왜곡(rate-distortion) 곡선을 도시한 것이다.

<181> 비트스트림의 구성을 설명한다. 압축된 비트스트림은 팔진트리의 노드데이터

로 구성된다. 비트스트림의 노드의 순서는 상술한 병합되는 노드의 역순으로 주어진다. 도 5는 노드의 비트스트림의 구조를 도시한 것으로서, 노드의 정보는 Continue 플래그, POSITION, SOP('S' 또는 'P'), DIB(detailed information bits) 데이터로 구성된다.

<182> 첫째, Continue 플래그는 압축된 비트스트림의 끝인지 여부를 나타낸다. 이들 모든 데이터는 적응산술코더(AAC)를 사용하여 엔트로피 부호화된다. 둘째, POSITION 데이터는 어떤 'B' 노드가 분리(split) 또는 PPM부호화되는가를 나타낸다. 예를 들어 디코더가 지금까지 수신된 도 3(d)에 도시된 근사화된 모델을 복원한다고 가정하자. 도 3(d)에 도시된 팔진트리에는 네 개의 'B'노드가 있으며, 도 3(c)에서와 같은 보다 정교한 모델을 얻기 위해서는 인코더는 디코더에게 왼쪽에서 두번째 'B'노드가 PPM 부호화되어야 함을 알려주어야 한다. 따라서 'B' 노드 후보 중 현재의 노드 순서는 POSITION 데이터로 부호화된다. 만일 한 개의 B 후보노드만 있다면, Position데이터는 필요없으며, 따라서 부호화되지 않는다. 셋째, SOP는 현재의 노드가 8개의 자식 노드들로 나누어져야 하는지 PPM-부호화되어야 하는지를 나타낸다. 만일 SOP가 현재노드는 'S'노드임을 나타내면, DIB는 현재 노드의 평균 컬러와 자식들 노드가 W노드인지 여부를 나타내는 8개의 플래그로 구성된다. W 노드가 아닌 노드는 일시적으로 B노드로 간주되며, 다음 비트스트림을 위한 B후보노드를 기록하고 있는 큐에 넣어진다. 도 6은 'S'노드의 구조를 도시하고 있다.

<183> 만일 SOP가 현재노드는 'P'노드임을 나타내면, 현재 노드 안에 있는 B 체적소의 깊이 정보가 PPM 부호화되며, B 체적소의 컬러정보는 DPCM 부호화된다. 도 7은 'P'노드의 비트스트림 구조를 도시한 것이다. PPM은 원래 팩시밀리 데이터를 무손실 방식으로 압축하기 위해 클리어리(Cleary)와 위튼(Witten)에 의해 제안된 것이다.

<184> PPM('P')노드의 부호화를 설명한다. 상술한 바와 같이 깊이 정보는 이진 체적소 값으로 표현된다. W 체적소는 투명한 배경을 표현하고, B 체적소는 3차원 객체를 나타낸다. P노드 내의 이진 체적소 값은 이웃하는 체적소 값을 컨텍스트로 사용하여 PPM 부호화된다. 도 8은 컨텍스트의 예를 도시한 것이다. 도 8(b)에 도시된 사각형 체적소가 부호화될 것이다. 사각형 체적소를 부호화하기 위해 13개의 원으로 된 체적소들이 컨텍스트로 사용된다. 래스터 스캔순서에 따라 W 체적소는 '0'으로 세팅되며, B 체적소는 '1'로 세팅된다. 이 13개의 비트는 사각형 체적소의 컨텍스트로 사용된다. 이 예에서, 사각형 체적소의 컨텍스트는 '0011101000011'이다. 그러나 이 예에서 컨텍스트의 수는  $2^{13}$  이다. 이것은 연산, 처리하기에 너무 크다. 따라서 컨텍스트의 수를 줄일 필요가 있다. 시뮬레이션에 의해 전체 엔트로피를 크게 증가시키지 않는 체적소의 제거를 선택한다. 먼저 두 개의 체적소가 제거된다. 그리고 나서 또 다른 체적소가 제거된다. 제거된 세 개의 체적소는 도 8(a)에 도시된 바와 같이 'X'로 마킹된다. 이 컨텍스트를 이용하여 체적소는 컨텍스트 기반 적응 산술코더(AAC)에 의해 압축된다.

<185> 도 8에서 (b)에 도시된 사각형 체적소가 부호화될 것이며, (a)와 (b)에 도시된 원으로 된 체적소는 컨텍스트로 사용된다. 그리고 도 8(a)에서는 제거된 체적소가 'X'로 마킹된 3 개의 체적소임을 보여준다.

<186> 깊이 정보를 부호화한 후 현재 노드 안에 있는 B 체적소의 R,G,B 컬러값은 래스터 스캔 순서로 DPCM 부호화된다. 보다 구체적으로 말하면, B 체적소의 R,G,B 값은 이전의 B 체적소 각각의 값들로부터 예측된다. 그리고 예측 오차(residual)는 산술 코더로 부호화된다.

<187> 상기 비트스트림 생성부(1130)는 노드 부호화된 데이터를 비트스트림으로 생성한다.

- <188> 상술한 본 발명은 3차원 객체 데이터의 부호화를 점진적으로 하는 점진적(progressive) 부호화 방식을 중심으로 설명하였지만, 무손실(lossless) 부호화 방식이나 손실(lossy) 부호화 방식으로 부호화할 수도 있다. 전체 비트스트림을 다 부호화하면 무손실 부호화 방식이 되고, 전체 비트스트림에서 처음부터 일부분까지만 부호화하면 손실 부호화 방식이 된다.
- <189> 다음은 3차원 객체 데이터의 복호화 방법 및 장치에 대해 설명하기로 한다.
- <190> 도 19는 3차원 객체 데이터의 복호화장치의 구성을 블록도로 도시한 것으로서, 비트스트림 독출부(1900), 노드복호화부(1910) 및 트리구조복원부(1920)을 포함하여 이루어진다. 상기 비트스트림독출부(1900)는 부호화된 3차원 객체 데이터의 비트스트림을 받아들이며, 상기 노드복호화부(1910)는 상기 비트스트림을 복호화하며, 상기 트리구조복원부(1920)는 상기 복호화된 노드를 트리구조로 복원한다.
- <191> 도 20은 상기 노드복호화부(1910)의 구성을 블록도로 도시한 것으로서, Continue 플래그 복호화부(2000), 노드위치정보 복호화부(2010), 노드종류선택부(2020), S노드복호화부(2030) 및 P노드복호화부(2040)를 포함하여 구성된다.
- <192> 상기 Continue 플래그 복호화부(2000)는 3차원 객체 데이터의 부호화된 비트스트림으로부터 Continue 플래그 정보를 읽어 복호화한다. 상기 노드위치정보 복호화부(2010)는 부호화된 3차원 객체 데이터의 비트스트림에 대해 노드위치의 값을 읽어 현재 노드가 큐에서 어느 후보노드인지를 나타내는 위치정보로 복호화한다. 노드종류선택부(2020)는 노드 종류 정보를 복호화한다. 상기 S노드복호화부(2030)는 상기 노드종류 정보가 S 노드를 나타내면, 상세정보의 값으로 하위 8개 노드들에 대한 평균 색상을 복호화하고 하위 8개 노드들을 차례대로 Black 또는 White로 복호화한다. 상기 P노드복호화부(2040)는 상기 노드종류 정보가 PPM 노드를 나

타내면, 상세정보값으로 현재 노드를 PPM 복호화하고 현재 노드의 B 체적소들의 색상인 R,G,B 값에 대해 역 AAC를 수행하여 복호화하고 역 DPCM를 수행하여 복호화한다.

<193> 도 21은 상기 노드복호화부(1910)의 동작을 흐름도로 도시한 것으로서, 먼저 Continue 플래그를 복호화하고(2100단계), 그 다음 Position 정보를 복호화한 후(2110단계), 노드가 P노드인지 검사하여(2120단계), P노드이면 P노드 복호화하고(2130단계), 그렇지 않으면 S노드 복호화한다.(2140단계) 상기 과정을 모든 모드에 대해 수행한다.(2150단계)

<194> 도 22는 상기 S노드복호화부(2020)의 동작을 흐름도로 도시한 것으로서, 평균 컬러정보를 복호화하고(2200단계), 8개의 하위노드 라벨을 복호화한다.(2210단계) 도 23은 상기 P노드 복호화부(2030)의 동작을 흐름도로 도시한 것으로서, PPM 깊이정보를 복호화하고(2300단계), 컬러정보를 역DPCM을 통해 복호화한다.(2310단계) 보다 구체적으로 설명하면 DIB값으로 현재 노드를 PPM 복호화하고 현재 노드의 Black 체적소들의 색상인 R,G,B 값을 역 AAC를 수행하여 복호화하고 역 DPCM를 수행하여 복호화한다.

<195> 이를 보다 상세히 설명하기로 한다. 디코딩 과정은 인코딩과 유사하다. 먼저 복호화기는 비트스트림의 Continue 플래그를 읽어서, 비트스트림의 끝인지 여부를 알아낸다. 그 다음 POSITION과 SOP 데이터를 읽어 디코더는 어떤 B 노드가 분리(Split)되어야 하는지 또는 PPM복호화 되어야 하는지를 찾아낸다. 'S'노드의 경우 현재 노드의 모든 체적소의 컬러 값은 일시적으로 DIB 내에서 R,G,B 컬러 값의 평균값으로 설정된다. 이 컬러 값들은 자식들(children) 데이터가 수신될 때 갱신된다. 'P'노드의 경우 현재 노드의 B 체적소의 깊이 및 컬러 정보는 DIB를 사용하여 손실없이 복원된다.

<196> 상술한 3차원 객체 데이터의 복호화는 점진적으로 복호화하는 점진적(progressive) 복호화방식을 중심으로 기술하였다. 즉 점진적 복호화방식을 사용하면 영상데이터의 비트스트림의

복호화되는 노드별로 복호화가 진행된다. 따라서 상기 복호화된 데이터가 화면에 디스플레이 될 경우, 최상위 노드부터 시작되어 하위노드로 복호화가 진행되어 노드별로 화면에 디스플레이된다. 결국 화면에 디스플레이되는 영상은 노드가 하위로 진행될수록 점차 세밀하고 명확하게 디스플레이되어 최하위 마지막 노드까지 복호화되면 가장 세밀하고 명확한 영상이 디스플레이 된다.

<197> 또한 상술한 3차원 객체 데이터의 복호화는 점진적(progressive) 복호화방식 위주로 기술하였으나 무손실(lossless) 복호화 방식 또는 손실(lossy) 복호화방식으로 복호화를 할 수 있다. 즉 전체 비트스트림을 다 복호화하면 무손실 복호화 방식이 되고, 전체 비트스트림에서 처음부터 특정 노드까지만 복호화하면 손실 복호화 방식이 된다.

<198> 본 발명은 또한 컴퓨터로 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다. 컴퓨터가 읽을 수 있는 기록매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피 디스크, 광데이터 저장장치 등이 있으며, 또한 캐리어 웨이브(예를 들어 인터넷을 통한 전송)의 형태로 구현되는 것도 포함한다. 또한 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터시스템에 분산되어 분산방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다.

<199> 이상에서 본 발명의 바람직한 실시예에 대해 도시하고 설명하였으나, 본 발명은 상술한 특정의 바람직한 실시예에 한정되지 아니하며, 청구범위에서 청구하는 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 기술분야에서 통상의 지식을 가진 자라면 누구든지 다양한 변형 실시가 가능한 것은 물론이고, 그와 같은 변경은 청구범위 기재의 범위 내에 있게 된다.

## 【발명의 효과】

- <200> 본 발명에 의한 3차원 객체 데이터의 부호화 및 복호화 방법의 성능을 도 9에 도시된 테스트 모델을 사용하여 평가하였다. 도 9(a)-(e)에 도시된 모델들은 해상도 256x256x256을 가지며, 도 9(f) (h)는 512x512x512 해상도를 가진다.
- <201> 도 10은 포인트텍스처 이미지 상에서의 WinZip과 본 발명에 따른 방법의 압축성능을 비교한 것을 도시하고 있다. 깊이정보에 대해, 제안된 알고리즘은 WinZip 의 파일 크기의 22-59 % 만 요구한다. 컬러 정보에 대해, 제안된 알고리즘은 또한 WinZip 의 파일 크기의 59-81 % 만 요구한다. 유일한 예외는 'Plane' 모델이다. 이것은 다양하지 않은 단지 몇 개의 색깔만 포함하며, 이러한 경우는 사전기반 코더인 WinZip 이 더 나은 성능을 나타낸다. 'Plane' 모델을 제외하고, 제안된 알고리즘은 깊이와 컬러 정보 둘다를 압축하기 위해 WinZip 의 파일 크기의 49-75 % 만 요구한다.
- <202> 본 발명에 의한 방법은 WinZip이 제공하지 못하는 점진적 전송과 디코딩을 지원한다는 점에서 의미 있다. 점진성(progressiveness)은 바람직하다. 왜냐하면 그것은 대역폭이 제한된 환경에서 비트스트림의 적응성(Adaptation)과 3차원 모델의 브라우징을 용이하게 해주기 때문이다.
- <203> 포인트 텍스처를 위한 점진적(progressive) 압축 방법을 본 발명에서 게시하였다. 그 각각의 픽셀은 컬러 벡터 뿐만 아니라 깊이 값과 연관되어 있다. 본 발명에 의한 방법은 PPM 방법을 팔진트리 구조로 통합하므로써 깊이 및 컬러 데이터 둘 다를 효율적으로 부호화한다. 시뮬레이션 결과는 본 발명에 의한 방법이 범용 Lempel-Ziv 코더인 WinZip 보다 훨씬 높은 압축율을 제공함을 보여주고 있다.



- <204> 본 발명에 의한 3차원 객체 데이터의 부호화 및 복호화 방법의 성능을 도 9에 도시된 테스트 모델을 사용하여 평가하였다. 도 9(a)-(e)에 도시된 모델들은 해상도 256x256x256을 가지며, 도 9(f) (h)는 512x512x512 해상도를 가진다.
- <205> 도 10은 포인트텍스처 이미지 상에서의 WinZip과 본 발명에 따른 방법의 압축성능을 비교한 것을 도시하고 있다. 깊이정보에 대해, 제안된 알고리즘은 WinZip의 파일 크기의 22-59% 만 요구한다. 컬러 정보에 대해, 제안된 알고리즘은 또한 WinZip의 파일 크기의 59-81% 만 요구한다. 유일한 예외는 'Plane' 모델이다. 이것은 몇 개의 색깔만 포함하며, 사전기반 코더인 WinZip이 더 나은 성능을 나타낸다. 'Plane' 모델을 제외하고, 제안된 알고리즘은 깊이와 컬러 정보 둘다를 압축하기 위해 WinZip의 파일 크기의 49-75% 만 요구한다.
- <206> 본 발명에 의한 방법은 WinZip이 제공하지 못하는 점진적 전송과 디코딩을 지원한다는 점에서 의미 있다. 점진성(progressiveness)은 바람직하다. 왜냐하면 그것은 대역폭이 제한된 환경에서 비트스트림의 적응성(Adaptation)과 3차원 모델의 브라우징을 용이하게 해주기 때문이다.
- <207> 포인트 텍스처를 위한 점진적(progressive) 압축 방법을 본 발명에서 게시하였다. 그 각각의 픽셀은 컬러 벡터 뿐만 아니라 깊이 값과 연관되어 있다. 본 발명에 의한 방법은 PPM 방법을 팔진트리 구조로 통합하므로써 깊이 및 컬러 데이터 둘 다를 효율적으로 부호화한다. 시뮬레이션 결과는 본 발명에 의한 방법이 범용 Lempel-Ziv 코더인 WinZip 보다 훨씬 높은 압축율을 제공함을 보여주고 있다.

**【특허청구범위】****【청구항 1】**

포인트텍스처, 체적소, 팔진트리구조의 데이터 중 어느 하나로 이루어진 3차원 객체 데이터에 대해 노드의 종류를 나타내는 라벨이 부여된 트리구조의 3차원 객체 데이터를 생성하는 단계;

상기 라벨이 부여된 트리구조의 3차원 객체 데이터의 노드를 부호화하는 노드 부호화단계; 및

상기 노드 부호화된 데이터를 비트스트림으로 생성하는 비트스트림 생성단계를 포함함을 특징으로 하는 3차원 객체 데이터 부호화 방법.

**【청구항 2】**

제1항에 있어서, 상기 라벨이 부여된 트리구조의 3차원 객체 데이터 생성은

3차원 바운딩 볼륨을 사용하여 체적소 데이터로 변환하여 객체가 존재하는 위치의 체적소와 객체가 존재하지 않는 위치의 체적소를, 라벨을 붙여 구분하여 표시함을 특징으로 하는 3차원 객체 데이터 부호화 방법.

**【청구항 3】**

제2항에 있어서, 상기 라벨을 붙여 체적소를 구분하여 표시함은

상기 3차원 객체 데이터를 나타내는 트리의 노드가 부노드로 구성될 때는 'S' 로 표시하고, 노드내의 부체적소가 모두 객체가 없는 체적소로 구성될 때는 'W'로 표시하고, 노드 내의 부체적소가 모두 객체가 있는 체적소로 구성될 때는 'B'로 표시하고, 노드 내의 부체적소 값이

PPM 알고리즘에 의해 부호화될 때는 'P'로 표시함을 특징으로 하는 3차원 객체 데이터 부호화 방법.

【청구항 4】

제1항에 있어서, 상기 노드 부호화단계는

노드가 S 노드인지 P 노드인지에 관한 정보를 부호화하는 단계; 및

상기 노드정보가 S 노드이면 S 노드의 상세 정보를 부호화하고, P 노드이면 P노드의 상세 정보를 부호화하는 단계를 포함함을 특징으로 하는 3차원 객체 데이터 부호화 방법.

【청구항 5】

제1항에 있어서, 상기 노드 부호화단계는

라벨이 부여된 트리구조의 3차원 객체 데이터의 시작 노드부터 특정 노드까지만 부호화함을 특징으로 하는 3차원 객체 데이터 부호화 방법.

【청구항 6】

제1항에 있어서, 상기 노드 부호화단계는

라벨이 부여된 트리구조의 3차원 객체 데이터의 모든 노드를 부호화함을 특징으로 하는 3차원 객체 데이터 부호화 방법.

【청구항 7】

포인트텍스처, 체적소, 트리구조의 데이터 중 어느 하나로 이루어진 3차원 객체 데이터에 대해 노드마다 라벨이 부착된 트리구조의 3차원 객체 데이터를 생성하는 트리구조생성단계;

상기 노드마다 라벨이 부여된 트리구조의 3차원 객체 데이터를 상기 라벨을 참조하여 병합하는 병합단계;

상기 병합된 노드를 부호화하는 노드부호화단계;

상기 노드 부호화된 데이터를 비트스트림으로 생성하는 비트스트림 생성단계; 및

상기 병합단계, 노드부호화단계 및 비트스트림생성단계를 상기 트리구조의 3차원 객체 데이터의 최상위 노드까지 반복하는 단계를 포함함을 특징으로 하는 3차원 객체 데이터 부호화 방법.

#### 【청구항 8】

제7항에 있어서, 상기 라벨이 부여된 트리구조의 3차원 객체 데이터를 생성은

트리의 노드가 하위노드를 가질 때는 'S' 로 표시하고, 노드 내의 체적소가 모두 객체가 없는 체적소로 구성될 때는 'W'로 표시하고, 노드 내의 체적소가 모두 객체가 있는 체적소로 구성될 때는 'B'로 표시하고, 노드 내의 체적소 값이 PPM 알고리즘에 의해 부호화될 때는 'P' 로 표시함을 특징으로 하는 3차원 객체 데이터 부호화 방법.

#### 【청구항 9】

제8항에 있어서, 상기 병합 단계는

노드의 라벨이 'P'인 노드와, 하위 노드의 라벨이 'W'와 'B'인 'S'노드를 병합할 후보 노드로 선택하는 단계;

상기 후보노드에 대해 병합 전후의 왜곡된 비트수의 차(??D)와 병합 전후의 비트수의 차(??R)의 비가 최소가 되는 노드를 최적 노드로 선택하는 단계;

상기 최적노드의 라벨을 'B' 로 표시하는 단계;

상기 후보노드에서 최적 노드로 선택된 노드를 제외하고 상기 후보노드를 갱신하는 단계를 포함함을 특징으로 하는 3차원 객체 데이터 부호화 방법.

【청구항 10】

제9항에 있어서, 상기 병합 전후의 왜곡된 비트수의 차(D)는

원래 모델 V와 왜곡측정으로서의 그것의 근사치 간의 해밍 거리를 사용하여 계산하며,

$$D = \sum_{x=1}^X \sum_{y=1}^Y \sum_{z=1}^Z |V(x,y,z) - \hat{V}(x,y,z)|$$

(여기서 X Y Z 는 모델의 해상도이다.)에 의해 계산됨을 특징으로 하는 3차원 객체 데이터 부호화 방법.

【청구항 11】

제8항 또는 제9항에 있어서, 상기 노드 부호화는

상기 후보노드가 큐에 존재하는지 여부를 나타내는 정보에 해당하는 Continue 플래그를 부호화하는 단계;

상기 후보노드를 저장하는 큐 내에서의 위치 정보를 나타내는 노드 위치정보를 부호화하는 단계;

노드가 S 노드인지 P노드인지에 관한 노드종류정보를 부호화하는 단계;

상기 노드종류정보가 S 노드이면 S 노드의 상세정보를 부호화하고, P노드이면 P노드의 상세 정보를 부호화하는 단계를 포함함을 특징으로 하는 3차원 객체 데이터 부호화 방법.

【청구항 12】

제11항에 있어서, 상기 S 노드 부호화는

색상정보의 평균값을 부호화하는 단계; 및

8개의 부(sub)노드의 라벨을 부호화하는 단계를 포함함을 특징으로 하는 3차원 객체 데이터 부호화 방법.

**【청구항 13】**

제11항에 있어서, 상기 P 노드 부호화는

깊이정보를 부호화하는 단계; 및

색상정보를 부호화하는 단계를 포함함을 특징으로 하는 3차원 객체 데이터 부호화 방법.

**【청구항 14】**

제13항에 있어서, 상기 깊이정보의 부호화는

소정 개수의 콘텍스트를 사용하여 해당 노드 이하의 전체 하위노드를 래스터 스캔순서로 PPM 부호화함을 특징으로 하는 3차원 객체 데이터 부호화 방법.

**【청구항 15】**

제13항에 있어서, 상기 색상정보 부호화는

현재 노드의 'B' 체적소들의 색상인 R(Red), G(Green), B(Blue) 값을 DPCM 부호화하고, AAC 부호화함을 특징으로 하는 3차원 객체 데이터 부호화 방법.

**【청구항 16】**

제7항에 있어서, 상기 노드부호화단계는

상기 병합된 노드의 최상위 노드부터 특정 노드까지만 부호화함을 특징으로 하는 3차원 객체 데이터 부호화 방법.

**【청구항 17】**

제7항에 있어서, 상기 노드 부호화단계는

상기 병합된 노드의 모든 노드를 부호화함을 특징으로 하는 3차원 객체 데이터 부호화 방법.

**【청구항 18】**

포인트텍스처, 체적소, 트리구조의 데이터 중 어느 하나로 이루어진 3차원 객체 데이터에 대해 노드마다 노드의 종류를 나타내는 라벨이 부착된 트리구조의 3차원 객체 데이터를 생성하는 트리구조 생성부;

상기 트리구조의 3차원 객체 데이터에 대해 상기 라벨을 참조하여 병합하는 병합순서선택부;

상기 병합된 노드를 부호화하는 노드부호화부; 및

상기 노드 부호화된 데이터를 비트스트림으로 생성하는 비트스트림생성부를 포함함을 특징으로 하는 3차원 객체 데이터 부호화 장치.

**【청구항 19】**

제18항에 있어서, 상기 병합순서선택부는

노드의 라벨이 'P'인 노드와, 하위 노드의 라벨이 'W'와 'B'인 'S'노드를 병합할 후보노드로 선택하는 후보노드선택부;

상기 후보노드에 대해 병합 전후의 왜곡된 비트수의 차(??D)와 병합 전후의 비트수의 차(??R)의 비가 최소가 되는 노드를 최적 노드로 선택하여 상기 최적노드의 라벨을 'B'로 표시하는 최적노드선택부;

상기 후보노드 중 최적 노드로 선택된 노드를 제외하고 상기 후보노드를 갱신하는 후보 노드갱신부를 포함함을 특징으로 하는 3차원 객체 데이터 부호화 장치.

【청구항 20】

제18항 또는 제19항에 있어서, 상기 노드 부호화부는

압축된 비트스트림의 끝인지 여부를 나타내는 Continue 플래그를 부호화하는 Continue 플래그 부호화부;

상기 후보노드를 저장하는 후보노드 큐를 구비하고, 상기 큐 내에서의 위치 정보를 나타내는 노드 위치정보를 부호화하는 노드위치부호화부;

노드가 S 노드인지 P노드인지에 관한 노드종류정보를 부호화하는 노드 SOP 선택부;

상기 노드종류정보가 S 노드이면 S 노드의 상세정보를 부호화하는 S노드부호화부; 및

상기 노드종류 정보가 P 노드이면 P 노드의 상세정보를 부호화하는 P노드부호화부를 포함함을 특징으로 하는 3차원 객체 데이터 부호화 장치.

【청구항 21】

제18항에 있어서, 상기 노드부호화부는

상기 병합된 노드의 최상위 노드부터 특정 노드까지만 부호화함을 특징으로 하는 3차원 객체 데이터 부호화 장치.

【청구항 22】

제18항에 있어서, 상기 노드 부호화부는

상기 병합된 노드의 모든 노드를 부호화함을 특징으로 하는 3차원 객체 데이터 부호화 장치.



**【청구항 23】**

부호화된 3차원 객체 데이터의 비트스트림에서 Continue 플래그 정보를 읽어 복호화하는 단계;

부호화된 3차원 객체 데이터의 비트스트림에 대해 노드종류 정보를 복호화하는 단계;

상기 노드종류 정보가 S 노드를 나타내면 S 노드를 복호화하고, 상기 노드종류 정보가 PPM 노드를 나타내면 PPM 노드를 복호화하는 단계; 및

노드 복호화된 데이터를 트리구조로 복원하는 단계를 포함함을 특징으로 하는 3차원 객체 데이터 복호화 방법.

**【청구항 24】**

부호화된 3차원 객체 데이터의 비트스트림에 대해 노드 복호화하는 단계; 및

노드 복호화된 데이터를 트리구조로 복원하는 단계를 포함함을 특징으로 하는 3차원 객체 데이터 복호화 방법.

**【청구항 25】**

제24항에 있어서, 상기 노드 복호화단계는

부호화된 3차원 객체 데이터의 비트스트림에서 Continue 플래그를 읽어 이를 복호화하는 단계;

노드위치의 값을 읽어 현재 노드가 큐에서 어느 후보노드인지를 나타내는 위치정보로 복호화하는 단계;

노드 종류 정보를 복호화하는 단계;

상기 노드종류 정보가 S 노드를 나타내면, S 노드를 복호화하는 단계; 및

상기 노드종류 정보가 PPM 노드를 나타내면, PPM 노드를 복호화하는 단계를 포함함을 특징으로 하는 3차원 객체 데이터 복호화 방법.

【청구항 26】

제25항에 있어서, 상기 S 노드 복호화는

상세정보의 값으로 하위 8개 노드들에 대한 평균 색상을 복호화하고 하위 8개 노드들을 차례대로 B 또는 W로 복호화함을 특징으로 하는 3차원 객체 데이터 복호화 방법.

【청구항 27】

제26항에 있어서, 상기 PPM 노드를 복호화는

DIB값으로 현재 노드를 PPM 복호화하고 현재 노드의 Black 체적소들의 색상인 R,G,B 값을 역 AAC를 수행하여 복호화하고 역 DPCM를 수행하여 복호화함을 특징으로 하는 3차원 객체 데이터 복호화 방법.

【청구항 28】

부호화된 3차원 객체 데이터의 비트스트림을 받아들이는 비트스트림독출부;

상기 비트스트림을 복호화하는 노드복호화부; 및

상기 복호화된 노드를 트리구조로 복원하는 트리구조복원부를 포함함을 특징으로 하는 3차원 객체 데이터 복호화 장치.

【청구항 29】

제27항에 있어서, 상기 노드복호화부는

비트스트림이 계속되는 지를 나타내는 Continue 플래그를 복호화하는 Continue 플래그 복호화부;

부호화된 3차원 객체 데이터의 비트스트림에 대해 노드위치의 값을 읽어 현재 노드가 큐에서 어느 후보노드인지를 나타내는 위치정보로 복호화하는 노드위치정보 복호화부;

노드 종류 정보를 복호화하는 노드종류선택부;

상기 노드종류 정보가 S 노드를 나타내면, 상세정보의 값으로 하위 8개 노드들에 대한 평균 색상을 복호화하고 하위 8개 노드들을 차례대로 Black 또는 White로 복호화하는 S노드복호화부; 및

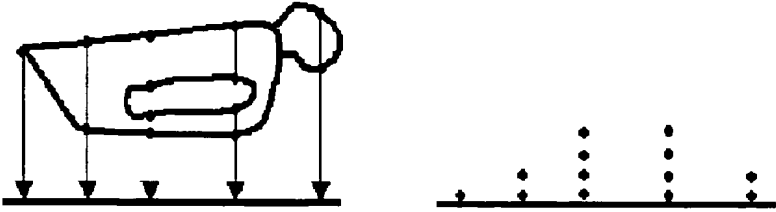
상기 노드종류 정보가 PPM 노드를 나타내면, 상세 정보 값으로 현재 노드를 PPM 복호화하고 현재 노드의 B 체적소들의 색상인 R,G,B 값에 대해 역 AAC를 수행하여 복호화하고 역 DPCM를 수행하여 복호화하는 P노드복호화부를 포함함을 특징으로 하는 3차원 객체 데이터 복호화 장치.

#### 【청구항 30】

제1항 내지 제17항, 제23항 내지 제27항 중 어느 한 항에 기재된 발명을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체.

【도면】

【도 1】

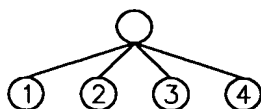


【도 2】

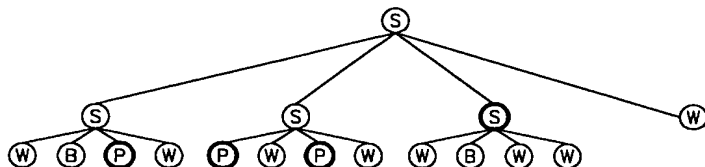
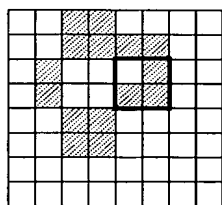
```
PointTexture {  
    field SFInt32    width      512  
    field SFInt32    height     512  
    field MFInt32    depth      []  
    field MFColor     color      []  
    field SFInt32    depthNbBits 8  
}  
to be compressed
```

【도 3】

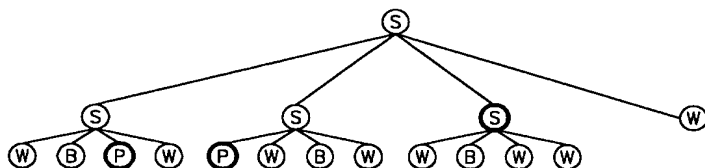
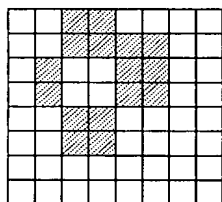
1	2
3	4



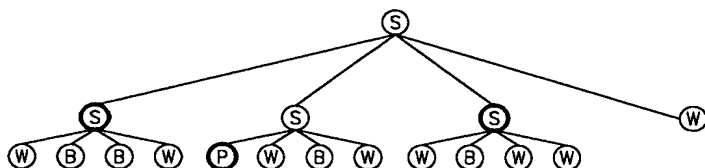
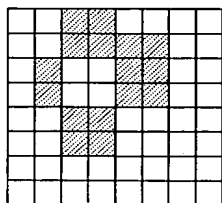
(a) 부모-자식관계(Parent-child relationship)



(b) 원래의 모델 및 상응하는 트리

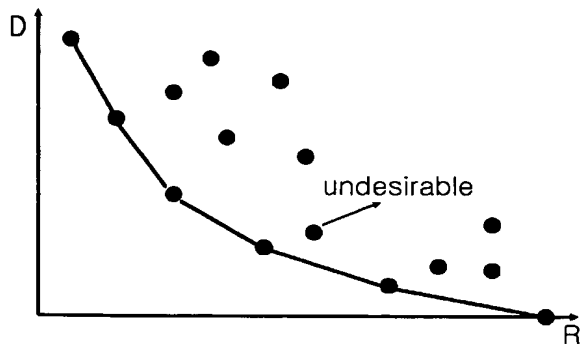


(c) 근사 모델 및 머징된 트리



(d) 근사 모델 및 머징된 트리

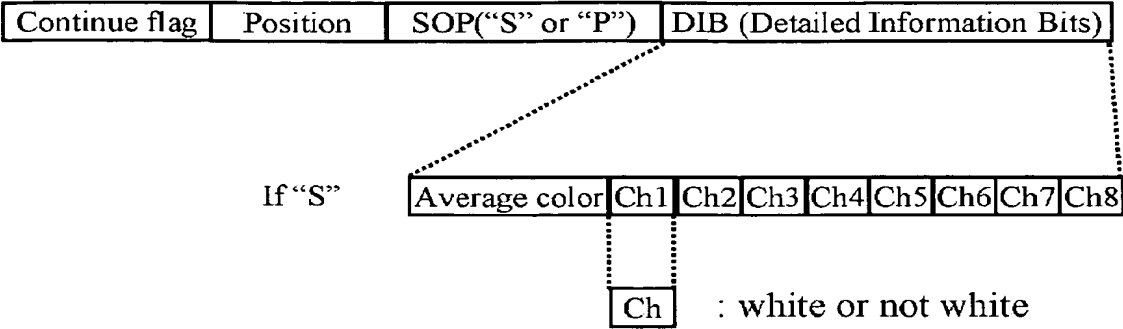
【도 4】



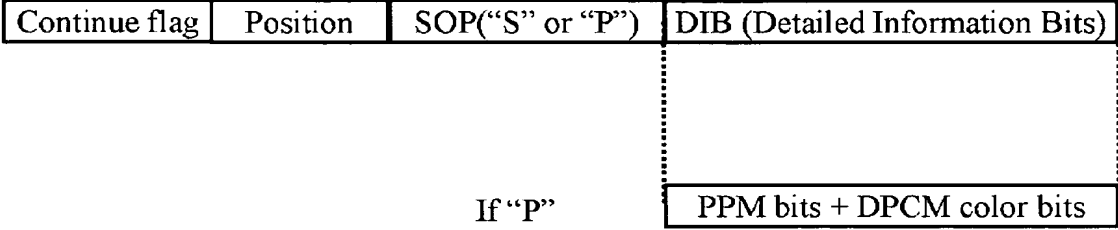
【도 5】

Continue flag	Position	SOP("S" or "P")	DIB (Detailed Information Bits)
---------------	----------	-----------------	---------------------------------

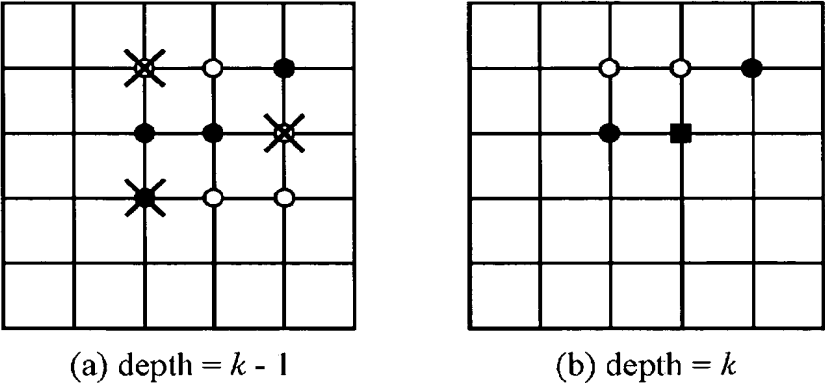
【도 6】



【도 7】



【도 8】



## 【도 9】

(a)~(e) resolution: 256x256x256 (f)~(h) resolution: 512x512x512



(a) angel



(b) avatar



(c) dog



(d) plane



(e) flower



(f) white star



(g) fighter



(h) robots

## 【도 10】

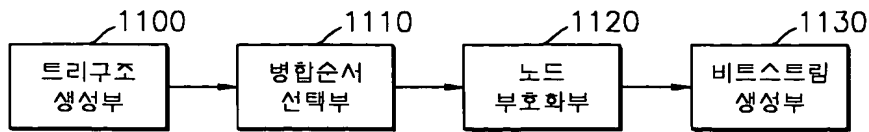
(a) : 원래의 이진 데이터와 비교

	DEPTH			COLOR			TOTAL		
	P	O	R	P	O	R	P	O	R
Angel	34,893	306,218	0.11	238,748	722,046	0.33	273,641	1,028,264	0.27
Avatar	6,396	87,184	0.07	35,813	64,944	0.55	42,209	152,128	0.28
Dog	11,501	111,262	0.10	64,312	137,178	0.47	75,813	248,440	0.31
Plane	5,254	96,946	0.05	51,256	94,230	0.54	56,510	191,176	0.30
Flower	17,804	98,469	0.18	65,016	98,799	0.66	82,820	197,268	0.42
WhiteStar	75,937	1,061,676	0.07	441,442	806,082	0.55	517,379	1,867,758	0.28
Fighter	38,108	906,308	0.04	275,853	573,030	0.48	313,961	1,479,338	0.21
Robots	105,458	1,285,188	0.08	669,236	1,141,350	0.59	774,694	2,426,538	0.32

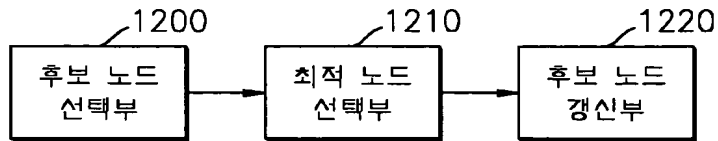
(b) : 상용 툴(WinZip)과의 비교

	DEPTH			COLOR			TOTAL		
	P	W	R	P	W	R	P	W	R
Angel	34,893	150,853	0.23	238,748	402,118	0.59	273,641	552,971	0.49
Avatar	6,396	14,407	0.44	35,813	46,475	0.77	42,209	60,882	0.69
Dog	11,501	34,256	0.34	64,312	112,734	0.57	75,813	146,990	0.52
Plane	5,254	9,771	0.54	51,256	47,245	1.08	56,510	57,016	0.99
Flower	17,804	30,243	0.59	65,016	80,721	0.81	82,820	110,964	0.75
WhiteStar	75,937	250,174	0.30	441,442	725,630	0.61	517,379	975,804	0.53
Fighter	38,108	173,124	0.22	275,853	414,241	0.67	313,961	587,365	0.53
Robots	105,458	330,815	0.32	669,236	939,136	0.71	774,694	1,269,951	0.61

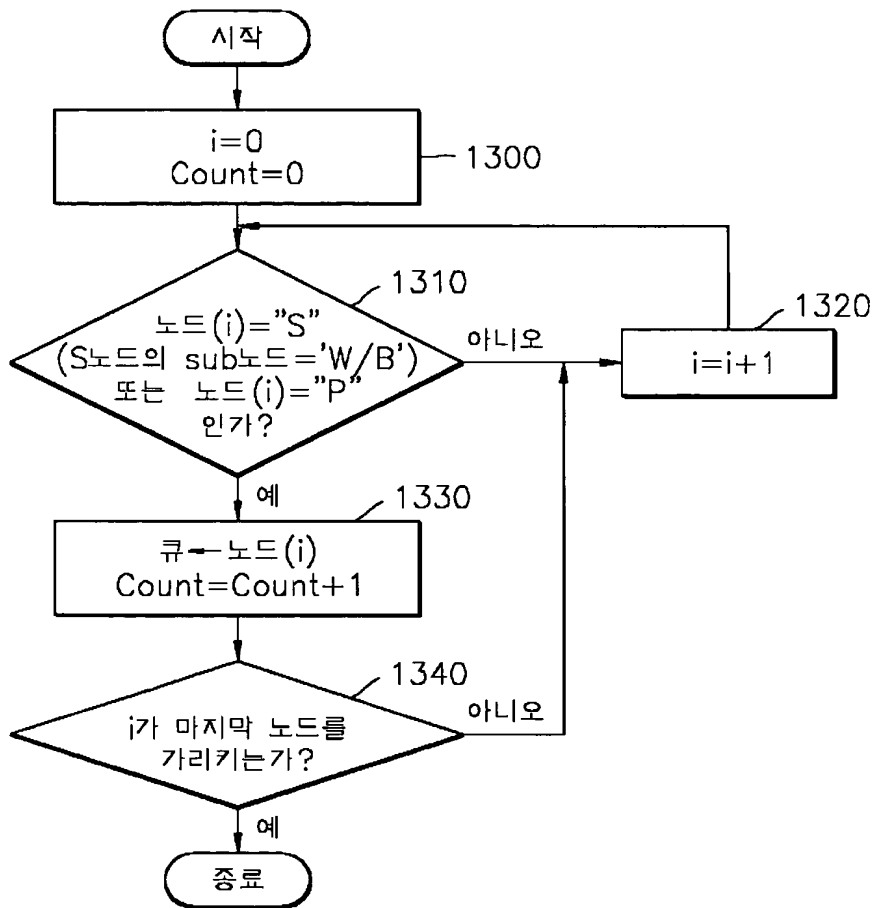
【도 11】



【도 12】

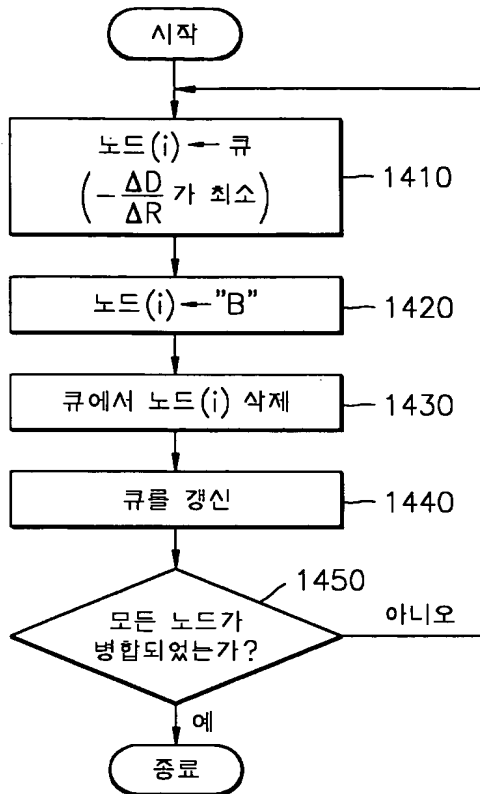


【도 13】

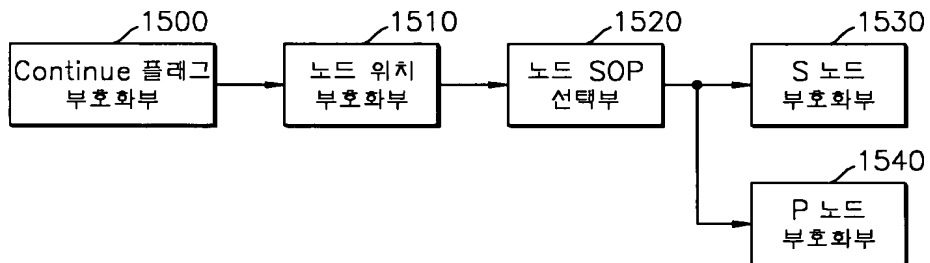




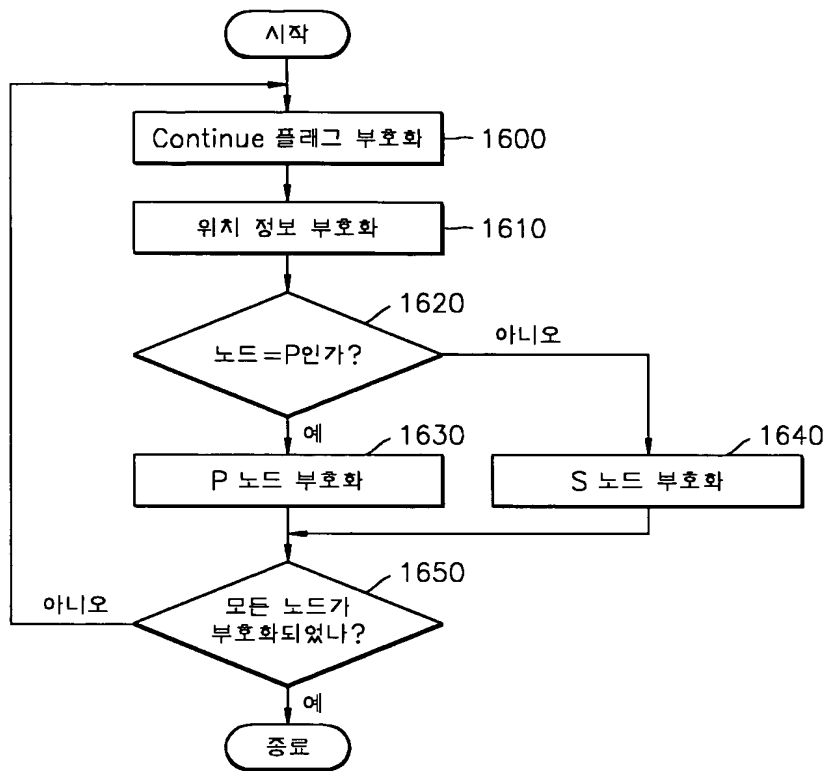
【도 14】



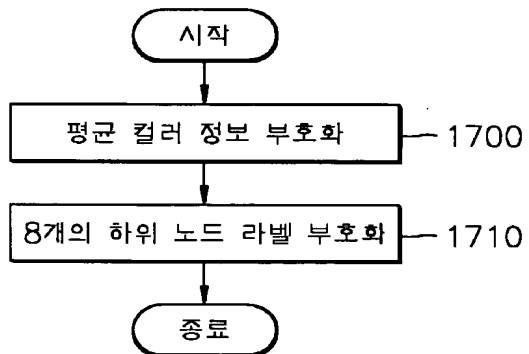
【도 15】



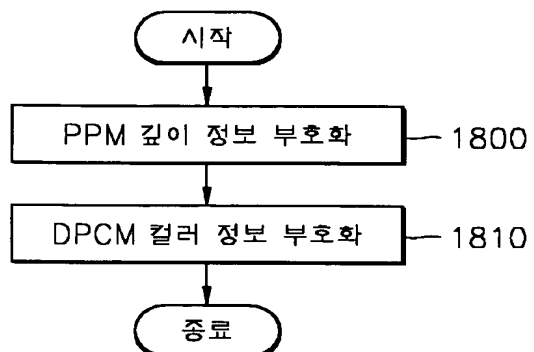
【도 16】



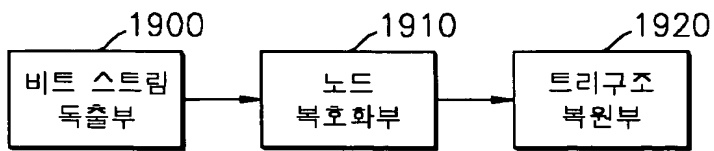
【도 17】



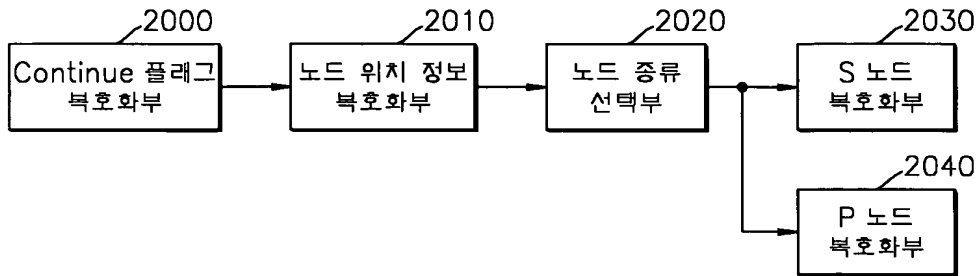
【도 18】



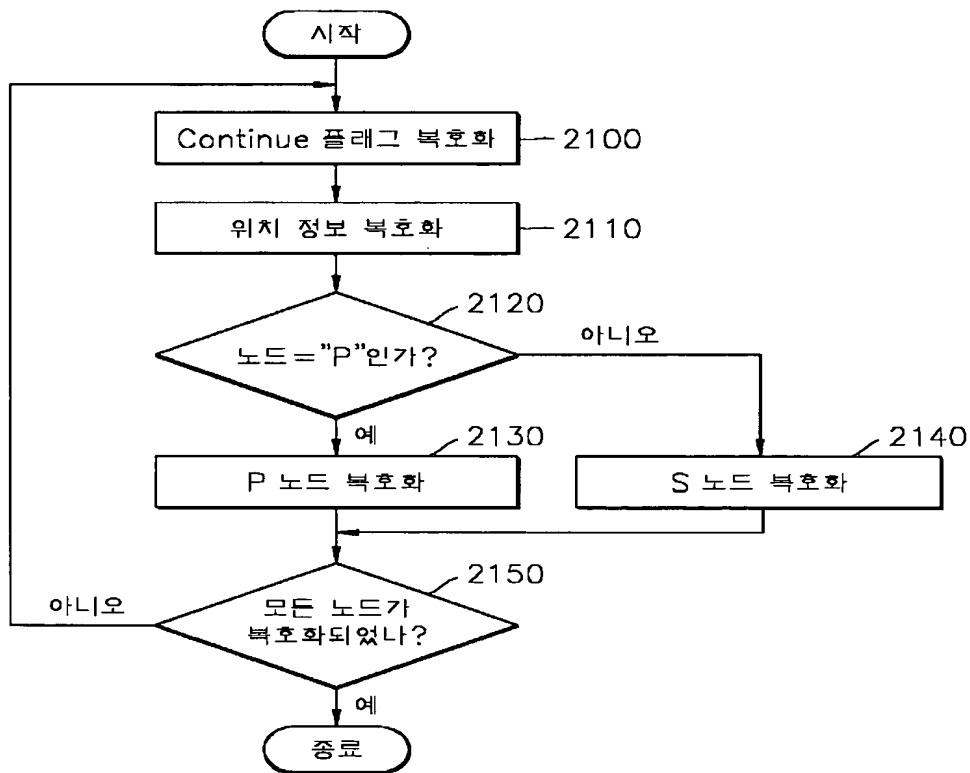
【도 19】



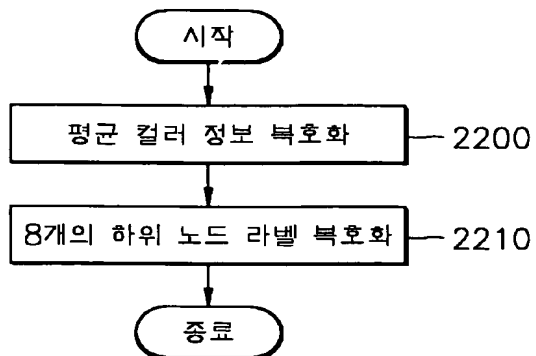
【도 20】



【도 21】



【도 22】



【도 23】

